

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра технічної кібернетики**

«На правах рукопису»  
УДК 004.0463

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Ігор ПАРХОМЕЙ

« \_\_\_\_ » \_\_\_\_\_ 2020 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-професійною програмою «Інформаційне забезпечення  
робототехнічних систем»**

**зі спеціальності 126 «Інформаційні системи та технології»**

**на тему: «Система моніторингу якості дорожнього покриття за  
допомогою мобільних пристроїв»**

Виконав :  
студент ІІ курсу, групи ІК-91мп  
Скрипка Даниїл Сергійович \_\_\_\_\_

Керівник:  
Доцент, к.т.н.,  
Польшакова Ольга Михайлівна \_\_\_\_\_

Консультант з нормоконтролю:  
доцент, к.т.н., доц.,  
Пасько Віктор Петрович \_\_\_\_\_

Рецензент:  
Старший викладач кафедри АУТС  
Могаль Олег Михайлович \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент (-ка) \_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра технічної кібернетики**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Ігор ПАРХОМЕЙ

«\_\_» \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ  
на магістерську дисертацію студенту**

**Скрипці Даниїлу Сергійовичу**

1. Тема дисертації «Система моніторингу якості дорожнього покриття за допомогою мобільних пристроїв», науковий керівник дисертації Польшакова Ольга Михайлівна доцент, к.т.н., затверджені наказом по університету від « 26 » жовтня 2020р. № 3132-с
2. Термін подання студентом дисертації \_\_\_\_\_
3. Об'єкт дослідження – Дорожнє покриття
4. Вихідні дані – відкриті дані інформаційних порталів, статті та підручники зарубіжних та вітчизняних авторів, існуючі прототипи програмних додатків.
5. Перелік завдань, які потрібно розробити – аналітичний огляд предметної області; огляд і порівняння існуючих рішень; дослідження ефективності автоматизації; розробка системи моніторингу якості дорожнього покриття.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу – п'ять креслень, один плакат
7. Орієнтовний перелік публікацій – 1 публікацій

## 8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перевірка на співпадіння	доцент Лісовиченко О.І.		
Нормоконтроль	доцент Пасько В.П.		

9. Дата видачі завдання \_\_\_\_\_

## Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області		
2	Аналіз існуючих прототипів та проблем		
3	Детальний опис предметної області		
4	Розробка алгоритмів програмного забезпечення		
5	Розробка інтерфейсу програмного забезпечення		
6	Маркетинговий аналіз стартап-проекту		
7	Попередній захист		
8	Нормоконтроль		
9	Перевірка на співпадіння		
10	Захист		

Студент

Даниїл СКРИПКА

Науковий керівник

Ольга ПОЛЬШАКОВА

**Пояснювальна записка  
до магістерської дисертації**

на тему: *Система моніторингу якості дорожнього покриття за  
допомогою мобільних пристроїв*

Київ – 2020 року



## АНОТАЦІЯ

У роботі розглянуто проблеми з дефектами на дорогах, які щоб зафіксувати потрібно затратити велику кількість ресурсів. Порівнянно п'ять різних систем моніторингу дорожнього покриття.

Розроблен алгоритм роботи системи, алгоритм модулю сегментації зображення, класифікації дефектів. Проведено огляд методів машинного навчання.

Розроблено інтерфейс мобільного додатку для системи моніторингу якості дорожнього покриття.

Ключові слова: система моніторингу, якість дорожнього покриття, сегментація, класифікація.

Розмір пояснювальної записки – 99 аркуша, містить 42 ілюстрації, 30 таблиць, 6 додатків.

## ABSTRACT

The problems with defects on the roads, such as the problem of fixing the need to spend a lot of resources. Five different systems are based on monitoring of road surface.

The algorithm of robotic systems, the algorithm for the image segmentation module, classification of defects have been broken up. An overview of the methods of machine learning is carried out.

The interface of the mobile add-on for the system and monitoring of the quality of the road surface has been broken.

Key words: monitoring system, quality of road coverage, segmentation, classification  
Explanatory note size – **99** pages, contains 42 illustrations, 30 tables, 6 applications.

## Зміст

СПИСОК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	10
ВСТУП.....	11
РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ.....	12
1.1 Об'єкт та предмет дослідження .....	12
1.2 Огляд існуючих рішень.....	12
1.3 Вибір картографічного рішення.....	17
1.4 Опис дефектів дорожнього покриття .....	18
1.5 Постановка задачі .....	21
Висновки до розділу.....	22
РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ .....	23
2.1 Аналіз методів машинного навчання .....	23
2.2 Google Maps API .....	30
2.3 Попередня обробка і витяг ознак в системах ВКДДП.....	32
2.3.1 Попередня обробка зображень в системах ВКДДП.....	32
2.3.2 Витяг ознак в системах ВКДДП.....	34
2.4 Математичне моделювання та чисельні методи сегментації в системах ВКДДП.....	36
2.4.1 Чисельний метод побудови карти дефектів дорожнього покриття .....	36
2.5 Математичне моделювання та чисельні методи класифікації дефектів .....	40
2.5.1 Проблема незбалансованих даних .....	40
2.5.2 Класифікація дефектів в системі ВКДДП.....	41
2.6 Застосування методів і алгоритмів ТЗ і МО для виявлення і класифікації об'єктів на зображеннях .....	45
2.7 Програмне середовище Swift 5 для розробки мобільної системи .....	45

Висновок до розділу .....	46
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ТА ЇЇ ПРОГРАМНА РЕАЛІЗАЦІЯ.....	48
3.1 Розробка структурної схеми системи .....	48
3.2 Розробка алгоритму роботи системи .....	49
3.3 Структура програми виявлення та класифікація дефектів в системах ВКДДП.....	50
3.3.1 Модуль введення / виводу даних .....	51
3.3.2 Модуль попередньої обробки зображення.....	53
3.3.3 Модуль сегментації зображення .....	54
3.4. Верифікація системи виявлення та класифікації дефектів дорожнього покриття.....	56
3.5 Розробка програмного забезпечення .....	58
3.6 Розробка інтерфейсу мобільного додатку .....	59
Висновки до розділу .....	62
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП – ПРОЕКТУ .....	63
4.1. Опис ідеї проекту.....	63
4.2. Технологічний аудит ідеї проекту .....	64
4.3. Аналіз ринкових можливостей запуску стартап – проекту .....	64
4.4. Розроблення ринкової стратегії проекту .....	70
4.5. Розроблення маркетингової програми стартап – проекту .....	71
Висновки по розділу .....	74
ВИСНОВКИ .....	75
ПЕРЕЛІК ПОСИЛАНЬ.....	77
ДОДАТКИ .....	78
ДОДАТОК А. ....	79

ДОДАТОК Б.....	80
ДОДАТОК В .....	81
ДОДАТОК Г.....	82
ДОДАТОК Ґ .....	83
ДОДАТОК Д .....	84
ДОДАТОК Е.....	85
ДОДАТОК Є .....	99

## **СПИСОК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ**

ШІ – Штучний інтелект

ІС – Інтелектуальна система

ЕОМ – Електронно – обчислювальна машина

МН – Машинне навчання

ЕС – Експертна система

DHDV – Digital Highway Data Vehicle

ВКДДП – Виявлення та класифікація дефектів дорожнього покриття

## ВСТУП

Автоматизація на основі штучного інтелекту відкриває безліч можливостей для діяльностей, які довго вважалися приналежністю виключно до людської сфери: розпізнавання закономірностей, прийняття рішень, обробка природної мови та візуальної інформації.

Для перетворення даних в корисні знання потрібно розвиток нових алгоритмів і програмних засобів. Таким чином, алгоритми машинного навчання (МН) стали областю популярних досліджень і додатків. МН – найважливіша область штучного інтелекту, яка має безліч реальних застосувань в різних областях.

На приклад автоматизація в автотранспортній галузі привертає велику увагу дослідників. Серед важливих чинників впливу на ефективність автодорожнього господарства відмітимо дефекти поверхні дороги. Дійсно, підтримка інфраструктури для надійного і енергоефективного функціонування автотранспорту є важливим завданням. Для утримання і планування ремонту доріг, дорожні компанії потребують точної і своєчасної інформації про дефекти доріг. У світі мільйони доріг, які необхідно перевіряти щороку. Раніше періодичні перевірки проводилися вручну інженерами.

Уже існують чимало систем, які використовують для аналізу доріг, проте жодна з цих систем не впроваджені в Україні через їх вартість та залежність від техніки. Таким чином було прийнято рішення розробити власну систему контролю якості, а щоб її можна було впровадити у систему контролю якості автомобільних доріг для дорожніх державних служб.

## РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

### 1.1 Об'єкт та предмет дослідження

**Об'єкт дослідження:** Дорожнє покриття.

**Предмет дослідження:** Система моніторингу якості дорожнього покриття.

**Мета дослідження:** Автоматизація, моніторинг та аналіз дорожнього покриття задля уникнення в подальшому дефектів на дорогах.

**Наукова новизна:** Створення доступної системи моніторингу якості дорожнього покриття для аналізу та реконструкції доріг для державних служб.

**Актуальність:** Неякісне дорожнє покриття на сьогоднішній день є крайне важливим, бо це сприяє великій загрозі, але своєчасно повідомити про це владу не є можливим без проведення аналізу. Головною метою роботи є створення системи моніторингу дорожнього покриття для служб дорожнього інспектування, щоб аналіз пошкоджень не вимагав великої кількості ресурсів.

### 1.2 Огляд існуючих рішень

Останніми роками отримали широке поширення цифрові системи. Такі системи обробляють відео потік досить високого розділення. На основі цього можна зробити висновок про необхідність залучення використання алгоритмів машинного навчання, що дозволяють обробляти великі масиви даних. Існує досить велика кількість систем автоматичного виявлення тріщин з використанням лінійних сканерів. Пропоновані системи проводять класифікацію згідно з шириною дефекту.

**ARAN.** Автоматичний аналізатор доріг Fugro (ARAN) – це ведуча у світі система збору даних про дороги і обстеження доріг, здатна вимірювати більше 15 різних потоків даних безперервно і з різною швидкістю збору даних за один прохід при швидкості руху.

Fugro ARAN – це вдосконалена машина для збору даних про дорожнє покриття, оснащена точними системами дорожнього обстеження і програмним забезпеченням, здатним виконувати будь – яке завдання збору даних про проїжджу частину. Модульна конструкція дозволяє легко переконфігурувати ARAN відповідно до вимог клієнтів, а також допомагає скоротити надлишки, оснащуючи ARAN тільки



необхідними геодезичними системами. У міру старіння автомобіля підсистеми можуть бути перенесені на інше шасі, що робить ARAN хорошою довгостроковою інвестицією.

За допомогою підсистеми візуалізації ARAN, вид стругання цифрові зображення (формат JPEG) записуються безпосередньо на диск для 100% проїжджої смуги (до 14 футів або 4,3 м). Цифрова сховище налаштовано для створення повного резервного копіювання кожного зібране зображення для захисту від втрати даних та уникнення часу споживаючи переробку.

Всі підсистеми візуалізації, пропоновані Fugro, включають штучне освітлення для забезпечення кращої консистенції та дозвільність зображень на тротуарі, що в свою чергу призводить до більшого точне визначення лиха. Завдяки архітектурі ARAN, зображення з видом на дорогу спеціально закодовано, щоб це було можливо подальший аналіз лиха за допомогою або напівавтоматизованих, або повністю автоматизоване програмне забезпечення для аналізу лиха.



Рисунок 1.1 ARAN автоматичний аналізатор доріг

Доступні конфігурації системи включають:

**Pave3D :** Оптимально система Pave3D збирає безперервний 3D зображення всіх типів дорожніх покриттів. Це унікальне тривимірне бачення технологія дозволяє точно вимірювати стан дорожнього покриття, вдень або вночі, до швидкості на шосе.

Роздільна здатність 2 мм із сильним освітленням: оптимально під кутом а синхронізовані з камерою стробоскопи забезпечують інтенсивний штучний освітлення навіть при наявності природного сонячного світла. Якість а інтенсивність світла, яку забезпечують стробоскопічні системи, перевершує що забезпечується стандартними системами розжарювання, які мають тенденцію вимагати зйомок під

час недостатнього освітлення доби, щоб забезпечити оптимальна якість зображення. Дозволяють високоінтенсивні стробоскопічні системи візуалізація тротуару протягом усіх годин доби та в усі дні умови освітлення (наприклад, похмуро, повне сонце тощо).

### **Компоненти системи**

Камери доступні з роздільною здатністю захоплення дрібних тріщин від ширини 2 мм (0,07 дюйма) до 1 мм (0,03 дюйма) на асфальтових поверхнях із гарячою сумішшю.

Фотоапарати та зняті зображення контролюються система цифрового відеозберігання (DVS). Через графічний DVS коригування зображення в інтерфейсі користувача (GUI) може бути зроблено до або під час збору.

### **Точність**

За допомогою ARGO DMI та GPS компанії Fugro повідомляється про місцезнаходження зображення в межах 10 см (3,94 дюйма) від фактичного геопросторового положення.

### **Вихідні дані**

Усі зображення зберігаються у стисненому jpeg без втрат формат. Метадані, включаючи інформацію про локатор, зберігаються в зображення та може бути відео у верхній частині банерного зображення

**WayLink Digital Highway Data Vehicle.** Цифровий транспортний Засіб Даних (DHDV) Шосе – система огляду шосе, що розвивається дослідниками в Університеті Арканзаса починаючи з 1990 – х. Фігура 1 показ зовнішня поява третього і поточного покоління DHDV . DHDV багато – функціональний і включає під – систему для відображення поверхні тротуару, під – системи для відображення (РЯД) шляху, і під – системи для дорожньої грубості і борозниць вимір. Інше технічне забезпечення включає одержувача GPS, Інструмент Виміру (DMI) Відстані, джерело живлення, і датчик Gyro. Система програмного забезпечення використовувала у бортових комп'ютерах DHDV наймає в реальному часі реляційний механізм субд, міжкомп'ютерна техніка комунікації, багато – комп'ютер і multi – сри базували паралельні обчислення, управління в реальному часі дискретних датчиків, і покоління

мультимедійних баз даних. У оглядах злітно – посадочної смуги в Атланті, тільки тротуар, що зображує підсистему, був використаний. Він може захопити зображення тротуару з тим, більш ніж 13 – футова ширина. Системи від другого і третього покоління DHDV отримують роздільну здатність 1mm/пікселя як подовжній брус, так і напрями в швидкості понад 100 КРН (близько 60mph). Зображений день системних робіт або ніч і це інваріантно до різної погодної умови із – за його використання лазера базував освітлення в невидимий вузький спектр.

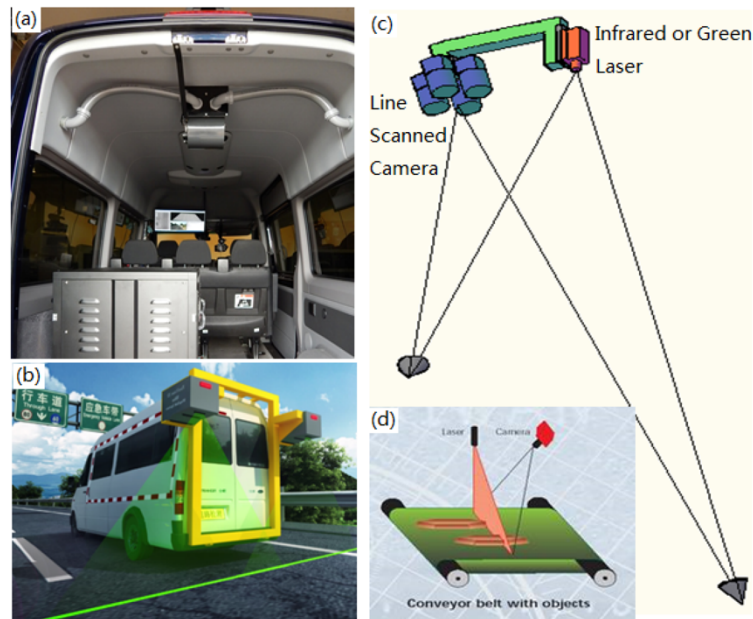


Рисунок 1.2 WayLink Digital Highway Data Vehicle

Два види цифрових фотоапаратів зазвичай використані в системі збору даних в тротуарі: камери рядкової розгортки і область переглядають камери. Перший покоління DHDV користувався камерою сканування області. Другий покоління DHDV користувався камерою рядкової розгортки. Обое з них користувалися традиційними видимими легкими джерелами для освітлення. Третій а також поточний покоління DHDV користується Лазерною Дорожньою Системою (Lris) формування або обробки зображень, яка наймає камери рядкової розгортки і лазерне освітлення. Третій покоління DHDV також забезпечує кращу якість зображення із – за його одноманітності освітлення, незважаючи на факт, що другий і треті покоління DHDV мають той же оптичний дозвіл. На (рис 1.3) показано робочий принцип для лінії переглядають камеру.

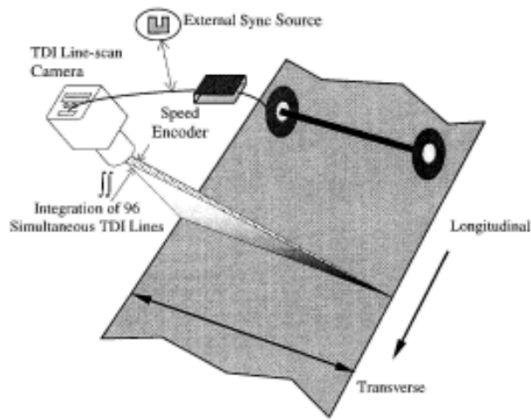


Рисунок 1.3 Камера Рядкової Розгортки.

**PAVUE.** У європейських країнах система знайшла застосування в Нідерландах і Фінляндії. Система оснащена двома відеокамерами для збору даних. Система використовує високошвидкісну технологію обробки, що об'єднує пристрій для збору і зберігання даних. Цей пристрій включає цифрові камери, ультразвукові і лазерні технології для збору структурних дефектів на дорозі.

**IRIS.** складається з камер високого розділення і лазерних джерел. Ці камери встановлені на обладнанні для збору цих автомобілях для і класифікації дефектів дорожнього покриття. Дані прочитуються в автономному режимі за допомогою автоматичної програми виявлення тріщин. Перевагами системи є швидкість і безпека.

**GIE LaserVISION** є яскравим прикладом використання лазерної технології для автоматичного виявлення і класифікації дефектів дорожнього покриття. Система використовує чотири лазерні датчики і забезпечує 3D – обробку для поліпшення розрахунку дефектів в 3D. Проте, дозвіл системи низький, і працює тільки з горизонтальними тріщинами. Основні проблеми обробки відеоданих подібними системами пов'язані з освітленням, кутом реєстрації зображень дорожнього полотна та ін. Щоб зменшити вплив цих чинників, система StereoVision використовує 3D – технологію для підвищення продуктивності системи.

Обчислювальна система Lvs складається з одноплатного комп'ютера – , Pentium ПК оперативної ПАМ'ЯТІ/32 – мегабайтів 200 – мгц підвищився в шасі градуса – – промисловця від Індустріального Комп'ютерного Джерела. ПК служить головним процесором і управляє усіма інтегральними технологіями придбання даних розмістив

на правлінні розглядаючий транспортний засіб. Ці технології включають 3D, вішення для огляду зовнішнього лиха, уклономірів для градусу і crossfall вимірів, глобальної системи global для географічних координат, і відстані, що вимірює інструмент для лінійного розташування. 3D, вішення технології, управляється матрицею трансп'ютерів 29 процесори T805 Sgs Thompson. Ці процесори виконують функції придбання даних для шести датчики Viris і обробка даних в реальному зволікаєте придбаних даних. Обробка даних в реальному зволікаєте займається торгівлею з чотирма main functions: це виправляє придбані профільні' позиції, зшиває профілі для повних поперечних і подовжніх профілів, фільтрує профілі і detects.

Таким чином, проаналізувавши існуючі системи, можна виявити, що кожна з наведених систем має велику точність та багатий функціонал. Проте це дуже великі та дорогі системи, які вимагають дуже багато ресурсів. Жодна з систем не може запропонувати «мобільну» версію, якою можна користуватися без виклику великої техніки.

### **1.3 Вибір картографічного рішення**

Наступною проблемою при створенні системи моніторингу якості дорожнього покриття стала проблема вибору картографічного сервіса. Було вирішено вибрати один з наступних картографічних сервісів:

Google Maps – набір додатків, побудованих на основі безкоштовного картографічного сервісу і технології, запропонованих компанією Google. Створено в 2005 році. Будь-які об'єкти непогано видно при збільшені масштабу карти. Є можливість прокласти маршрут для автомобілів, пішоходів та громадському транспорту. Також розраховується час переміщення цим маршрутом та інструкція як дістатися точки. Щоб використовувати потрібно зареєструватись, потім видається API key. Безкоштовне використання у некомерційних проектах.

2GIS – Карта країн СНД . Будь-які об'єкти непогано видно при збільшені масштабу карти. Є можливість прокласти маршрут для автомобілів, пішоходів та громадського транспорту транспорту. Також розраховується час переміщення цим маршрутом та інструкція як дістатися точки. Щоб використовувати потрібно

зарегіструватися, потім видається API key. Безкоштовне використання у некомерційних проектах.

Yahoo! Maps – безкоштовний картографічний сервіс, надавати пріоритет –ється компанією Yahoo !. Картографічні дані представлені NAVTEQ і Tele Atlas. Високо детальні космознімки метрового розрешення представлені Yahoo! Maps компанією GeoEye.

Maps.me – безкоштовний додаток для мобільних пристроїв на основі вільної географічної карти OpenStreetMap. З листопада 2014 року Maps.me належить Mail.ru Group і розвивається в рамках її міжнародного проекту My.com. Bing Maps – картографічний сервіс Microsoft, частина порталу Bing.

Було вирішено вибрати картографічний сервіс Google Maps. Google Maps це безкоштовна служба, яка не містить реклами, хоча така можливість зарезервована на майбутнє щодо їх використання. В Google Maps доступно 20 рівнів масштабу, що в сумі з одним з кращих покриттів в світі дає дуже хорошу деталізацію по всьому світу. Google Maps легкий і зручний в застосуванні, до нього написана детальна документація і велика кількість прикладів, які добре демонструють способи використання Google Maps. Але все ж основний критерій вибору Google Maps – наявність Google Maps API, аналога якому немає у інших картографічних сервісів. Google Maps API дозволяє робити нативні програми, які є більш швидкі – ми і менш ресурсоємними в порівнянні з гібридними додатками.

#### **1.4 Опис дефектів дорожнього покриття**




Дефекти дорожнього покриття залежать від багатьох причин як антропогенного, так природного характеру: клімат, транспорт, тривале використання в часі, якість матеріалів і так далі. Таким чином, природа дефектів може бути найрізноманітнішою. У різних країнах проводилися досить великі дослідження спрямовані на автоматизацію розпізнавання і класифікації дефектів дорожнього покриття.

Розрізняють дефекти цементобетонного і асфальтобетонного покриттів, які в свою чергу розділяються на лінійного (тріщини, руйнування країв дорожнього одягу,

руйнування деформаційних швів, колійність, зміщення) і площадного (сітка тріщин, вибоїни, викришування, лушення, просідання, проломи) характеру.

Проте у даній системі будуть розглядатися лише 3 види дефектів : глибокі тріщини, мережа (сітка) тріщин, вибоїни.

Таблиця 1.1 Опис основних дефектів дорожнього покриття

Тип дефекта	Опис	Приклади
Глибокі тріщини	– глибокі тріщини зазвичай зустрічаються в паралельних областях дорожнього покриття. Положення коліс автомобіля чинить пряму дію на поверхню і появу таких дефектів.	
Мережа тріщин	– тип дефекту покриття, який розвивається з горизонтальних тріщин і вертикальних тріщин. Цей дефект зазвичай з'являється на великих ділянках поверхні	
Вибоїна	– це ушкодження дорожньої поверхні, яка піддається дії, декількох шарів покриття тому руйнуються.	

Зображення дефектів дорожнього покриття – основний тип даних системи виявлення і класифікації, яка запропонована в цій роботі. Дані заздалегідь обробляються, з них витягаються ознаки, потім формуються вектори ознак. У реальних завданнях початкові дані відомі з точністю до рівня шуму. Фактичні умови довкілля (світло, дорожні характеристики і так далі) роблять великий вплив на збір даних. Підхід до рішення задачі полягає у виборі алгоритмів, налаштування їх параметрів, комбінацій методів обробки.

Дефекти покриття є об'єктами вхідних зображень системи виявлення і класифікації дефектів. Витягання ознак об'єкту є важливим кроком в аналізі даних. Для цього в роботі передбачається використання методів обробки зображень для аналізу і перетворення інформації зображення у вектори ознак для виявлення і класифікації дефектів дорожнього покриття.

Для вирішення завдання витягання ознак дефектів дорожнього покриття на основі цих зображень і відео застосовувалися методи з використанням вейвлет – перетворення, метод вибірки, міра анізотропії, метод на основі алгоритмів нечіткої логіки, метод вибіркової екстракції. У представлений метод витягання з використанням вейвлет – преобразовання. Використання вейвлет – перетворення дозволяє проводити аналіз дефектів тріщин на дорожньому покритті. На рис(1.4) описаний процес декомпозиції зображення на основі вейвлет – преобразовання, де  $L$  і  $H$  відповідно високочастотні.

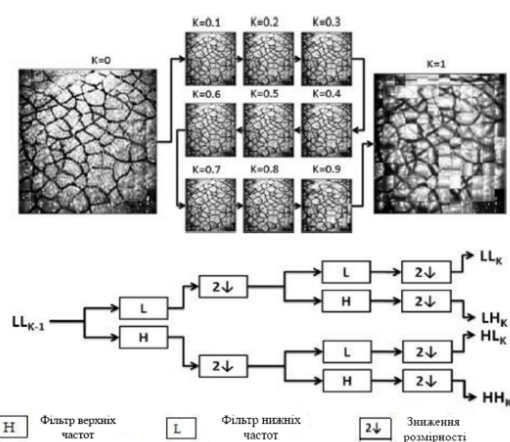


Рисунок 1.3. Метод витягання ознак дефектів дорожнього покриття методом Wavelet – Random transform



Аналіз форми дуже важливий для виділення дефектів. У дослідженні використовувався метод витягання лінійних ознак глибоких тріщин на дорожньому покритті – лінійного зносу. Термін "лінійний знос" уперше був запропонований в роботі [1]. На рис.1.3 описується використання двовимірного дискретного вейвлет – перетворення і фільтрів на основі методів математичної морфології.

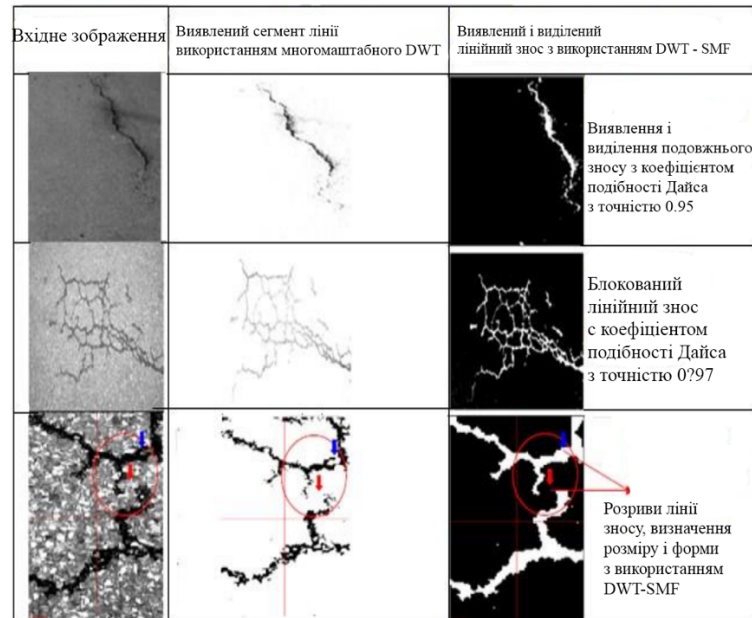


Рисунок 1.4. Метод витягання ознак на основі DWT – SMF (discrete wavelet transform – successive morphologic transform filtering)

## 1.5 Постановка задачі

Основними задачами магістерської дисертації є:

- Аналіз ефективності моніторингу якості;
- Дослідження підходів проектування системи;
- Дослідження підходу інтегрування машинного навчання;
- Розробка модулів аналізу зображення;
- Розробка алгоритму обробки вхідних даних (дорожнього покриття);
- Розробка структури системи;
- Розробка мобільного інтерфейсу згідно нижчеперерахованим вимогам.

Графічний інтерфейс – це спосіб взаємодії користувача із даними, тому його розробка має бути одним із найважливіших аспектів розробки якісної системи.

Отже, програмне середовище повинно відповідати таким вимогам:

- Легкість в освоєнні та застосуванні;
- Користувач повинен контролювати дії додатку;
- Візуальне сприйняття додатку не повинно відштовхувати користувача;
- Інтуїтивно – зрозумілий інтерфейс;
- Наявність функцій збереження та завантаження налаштувань;

### **Висновки до розділу**

У розділі визначено предмет та об'єкт дослідження, детально розглянуто актуальність та описано новизну продукту. Проведено огляд існуючих рішень на ринку. На основі цієї інформації поставлено задачу дисертації та було визначено вимоги до системи, що буде розроблена. Система повинна враховувати недоліки розглянутих рішень.

## РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Аналіз методів машинного навчання

**Поняття МН.** Машинне навчання – великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися на ретроспективних даних. МН – це сімейство методів створення математичних моделей, що використовують навчання по прецедентах або індуктивне навчання.

Область МН за своєю суттю спрямована саме на рішення практичних завдань, пов'язаних з обробкою великих масивів даних. МО, будучи міждисциплінарною областю, спирається на багато розділів класичної математики – передусім математичну статистику, методи оптимізації і дослідження операцій.

Нині алгоритми МН застосовуються для створення важливих застосувань в усіх сферах життя. Наприклад, Google Brain був розроблений в 2011 році для вирішення завдань штучного інтелекту. У 2015 році Microsoft створила відкриту платформу "Distributed Machine Learning Toolkit", яка дозволяє ефективно розподіляти рішення завдань МО на декількох комп'ютерах.

Впровадження методів МН дозволяє підвищити ефективність роботи багатьох застосувань і пристроїв. Якщо раніше МН вимагало складного програмного забезпечення, передових комп'ютерних систем і досвідчених учених, щоб настроїти і впровадити методи МН, то тепер розроблені високоефективні алгоритми МН, які простіше налаштовувати і застосовувати в конкретних областях науки і техніки.

**Типи завдань МН.** Існують 3 основні типи завдань МН: навчання з учителем, навчання без учителя, навчання з підкріпленням. Навчання з підкріпленням – це поліпшений тип моделі навчання з учителем. Навчання з учителем: Це метод МН припускає наявність розмічених даних. Є безліч об'єктів і безліч можливих міток. Існує деяка залежність між мітками і об'єктами, але вона невідома. Відома тільки кінцева сукупність прецедентів – пара "об'єкт, мітка", що називається повчальною вибіркою.

Завданням навчання з учителем є прогнозування виведення на основі вхідного значення. На основі цих даних вимагається відновити залежність, тобто побудувати алгоритм, здатний для будь – якого об'єкту видати досить точну відповідь. Тобто

зіставити об'єкту потрібну мітку віднісши його до певного класу (наприклад – дефект це або не дефект) в завданні класифікації або скаляр (вектор) в завданні регресії або прогнозування. Навчання з учителем застосовується до основних завдань класифікації і регресії, наприклад: прогноз цін на акції, прогнози погоди; розпізнавання рукописного тексту.

Навчання без учителя: це метод МН, який вивчає широкий клас завдань обробки даних, в яких відомі тільки описи безлічі об'єктів (повчальної вибірки), і вимагається виявити внутрішні взаємозв'язки, залежності, закономірності, існуючі між об'єктами. Типовим прикладом навчання без учителя є завдання кластеризації. Навчання з підкріпленням – це поліпшений тип моделі навчання з учителем. Цей метод полягає в тому, що система вчитиметься в середовищі, взаємодіятиме з нею і отримуватиме винагороду за виконання дій. Навчання без учителя є найбільш поширеним застосуванням кластеризації, пошуку правил асоціації, заповнення пропущених і т.д.

**Методи МН і практичне застосування.** МН широко застосовується у багатьох сферах бізнесу і життєдіяльності: маркетингу, фінансах, банківській справі, страхуванні, науці, охороні здоров'я, безпеці, Інтернеті і так далі. Багато організацій і компанії у світі застосовують методи МН для максимізації прибутку і мінімізації витрат. У останні час МН активно застосовується для інтелектуального аналізу даних і розпізнавання образів.

Інтелектуальний аналіз даних. У цих роботах алгоритми МН застосовані до бази даних для виявлення правил і закономірностей. Інтелектуальний аналіз даних це сучасна концепція аналізу даних, що припускає, що дані можуть бути неточними, неповними, суперечливими, різнорідними, при цьому мати велетенські об'єми. Алгоритми аналізу даних можуть навчатися по прецедентах, тобто робити загальні висновки на основі приватних спостережень переробляючи початкові дані в інформацію, а інформацію в знання автоматично. При інтелектуальному аналізі даних велику роль грають методи кластеризації, що дозволяють шукати нові закономірності в даних великої розмірності.

Розпізнавання образів: це віднесення початкових даних до певного класу за допомогою виділення істотних ознак, що характеризують ці дані, із загальної маси несуттєвих даних для вирішення конкретного завдання. Розпізнавання образів відрізняється від штучного інтелекту в початковому використанні математичної мови і моделі, а не шляхом експерименту, замінивши експеримент логічними міркуваннями і математичними доказами.

Розглянемо класичну постановку завдання розпізнавання образів. Дана безліч об'єктів. Відносно них необхідно провести класифікацію. Множина представлена підмножинами, які називаються класами. Задані: інформація про класи, опис усієї великої кількості і опис інформації про об'єкт, приналежність якого до певного класу невідома. Вимагається за наявною інформацією про класи і опис об'єкту встановити до якого класу відноситься цей об'єкт.

**Методи МО в завданнях класифікації.** Класифікація – це одне з основних завдань МО, яке сортує об'єкти згідно приналежності зумовленим класам. Формалізація класів проводиться на попередньому етапі. Об'єкти класифікуються по класах на основі значень витягнутих ознак аналізованих об'єктів. Типові алгоритми класифікації включають: нейронні мережі, у тому числі методи глибокого навчання, дерева рішень, байесовские мережі, метод опорних векторів. Простим ефективним методом класифікації і регресії є метод  $k$  найближчих сусідів (KNN).

Усі ці підходи створюють моделі, здатні класифікувати невідомі нові об'єкти на основі цих аналогічних об'єктів, по яких заздалегідь проводилося навчання. Зображення і відео містять великий об'єм погано структурованої інформації. Тому потрібний глибокий попередній інтелектуальний аналіз даних для створення різних застосувань.

Рішення задачі класифікації проводиться в два етапи:

Етап 1: Навчання. Мета цього етапу – побудувати модель для визначення набору класів даних. Ця модель будується шляхом аналізу наборів повчальних даних, кожен з яких визначається значенням ознак. Кожен набір даних відноситься до одного із зумовлених класів. Набір даних аналізується для побудови моделі класифікації.

Етап 2: Тестування і оцінка. На цьому етапі використовується модель класифікації, отримана на кроці 1. Спочатку оцінюється точність моделі за допомогою тестових наборів. Ці набори вибираються незалежно від зразків, які були вивчені на кроці 1. Кількість правильно класифікованих об'єктів – це точність моделі класифікатора, ґрунтована на тестових даних.

**Комбінація методів класифікації.** Існує два підходи до вирішення цієї проблеми. Перший підхід – полягає в створенні кожного класифікатора незалежно один від одного, потім використовується метод голосування для вибору кінцевого результату об'єднання. Другий підхід – створення основних класифікаторів і привласнення значущості результатам кожного класифікатора.

Метод Бэггинга: використовує перший підхід. Процедура бэггинг показує високий приріст узагальнювальної здатності в порівнянні з алгоритмом, навченим за допомогою базового методу по початковій повчальній вибірці, в тих випадках, коли варіаційна складова помилки базового методу висока. До таких моделей відносяться зокрема вирішальні дерева і нейромережеві методи. При використанні як базовий метод вирішальних дерев процедура бэггинг призводить до побудови ансамблів вирішальних дерев (вирішальних лісів).

У цьому розділі представлені методи МО і ТЗ, розроблені математичні моделі і чисельні методи вирішення поставлених завдань. Робота сфокусована на вирішенні наступних завдань: поліпшення якості вхідних даних за допомогою попередньої обробки зображень, сегментація зображень, виявлення і класифікації дефектів. Запропонований підхід складається з чотирьох основних етапів (рис.2.1).

### **Методи МН у виявленні і класифікації дефектів дорожнього покриття**

Як вже було відмічено вище, методи МН досить широко застосовуються для вирішення завдання виявлення і класифікації дефектів дорожнього покриття. Ефективність методів МН оцінюються на основі наступних критеріїв: час виконання, стабільність системи для незбалансованих даних (наприклад, наявність рідкісних типів дефектів), стабільності системи відносно непрямих чинників (шум, світло і так далі), здатності інтерпретувати результати і використати мінімум параметрів для налаштування.



Рисунок 2.1 Загальна структура типових систем виявлення і класифікації дефектів дорожнього покриття

Методи і алгоритми МН адаптовані і застосовані для автоматичного виявлення і класифікації дефектів дорожнього покриття і мають загальну структуру, як показано на рис.(2.1). Ці алгоритми перераховані в таблиці 2.1. Розглянемо їх детальніше.

Таблиця 2.1 Аналіз алгоритмів МН

Тип машинного навчання	Навчання без вчителя	Навчання без вчителя	Навчання без вчителя	Навчання без вчителя
Точність	**	***	*	****
Швидкість класифікації	***	****	***	**
Середньоквадратичне відхилення з шумом	**	**	***	**
Поліпшення процесу навчання	**	****	****	**

**Дерева рішень.** Дерева рішень – це спосіб представлення правил в ієрархічній, послідовній структурі, де кожному об'єкту відповідає єдиний вузол, що дає рішення. Деревом рішень є деревовидна структура, в якій кожен вузол представляє відповідну ознаку. Кожна гілка представляє результат тесту. Вузли означають класи або

розподіли класів. Для класифікації невідомого зразка, застосовується метод дерев рішень.

Дерева рішень можна трансформувати в набір правил класифікації. Математична модель дерева рішень складається з алгоритму зверху "вниз", є прикладом "жадібного" алгоритму. Перевага цього методу полягає в тому, що його легко реалізувати, легко зрозуміти і має невисоку обчислювальну складність.

**Нейронні мережі.** Штучні нейронні мережі (ANN) характеризуються шарами, кожен з яких складається зі взаємозв'язаних вузлів, що містять функцію активації. ANN створюють систему обробки інформації, яка імітує роботу нейронів мозку людини. Обробка інформації в нервовій системі складається з двох основних частин: обробка вхідного сигналу і виведення вихідного сигналу. Ці два класи взаємодіють один з одним через один або 30 декілька прихованих шарів. Елементи в різних класах зважені. Вагове значення може бути виправлене шляхом навчання правилам через вхідні значення. Перевага цього підходу полягає в тому, що він є потужним методом моделювання, який був успішно застосований для класифікації дефектів дорожнього покриття.

**Наївний Байєсовський класифікатор.** Байєсовський метод класифікації ґрунтований на статистичних методах. Цей метод використовується в прогнозуванні вірогідності класів набору даних, які поміщені в окремі класи.

Байєсовская мережа дозволяє виражати стосунки між ознаками.

**Метод опорних векторів – МОВ** був уперше запропонований В. Н. Вапником в 1960 році для класифікації даних і притягнув увагу дослідників. МОВ – ефективний метод, який може бути застосований до широкого спектру завдань ідентифікації дефектів дорожнього покриття і класифікації. Метою методу МОВ є побудова вирішального правила шляхом проекції простору ознак в простір більш високої розмірності і побудову розділяючої гіперплощини (вирішального правила) з максимальним проміжком в цьому просторі ознак.

Класифікація складається з трьох основних частин: попередня обробка та поліпшення якості цифрових зображень, витяг геометричних об'єктів в кожній області зображення, виявлення та ідентифікація дефектів на основі



вейвлетпреобразований для зображень з низьким дозволом, використання медіанного фільтра для отримання порогових значень, використання морфологічних фільтрів для ознак форми, використання МО для класифікації типів дефектів дорожнього покриття.



Рисунок 2.2 Блок – схема обнаруження та ідентифікації дефектів на дорожньому покритті на цифрових зображеннях

На основі МН використовувався для виявлення і класифікації дефектів. Зображення сегментуються в регіони, витягуються форми і текстурні ознаки для виявлення вибоїн.

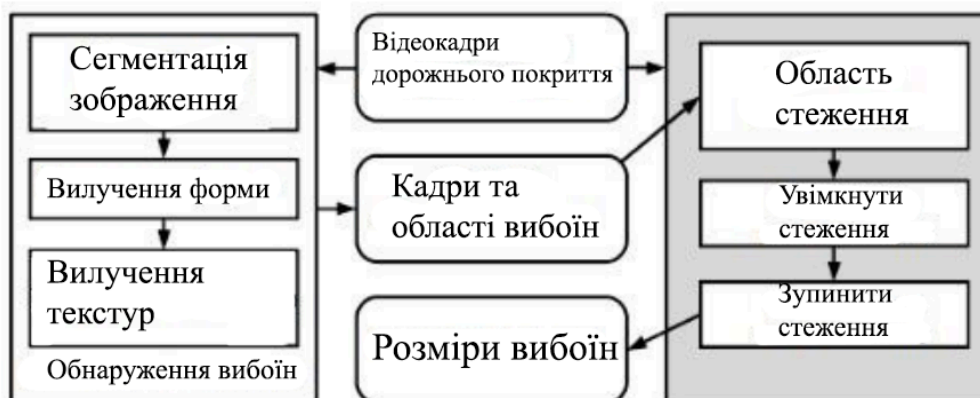


Рисунок 2.3 Опис основних частин системи пошуку дефектів вибоїн на відео

Система автоматичного виявлення і класифікації дефектів дорожнього покриття на основі комбінації вейвлет – перетворення і нейронної мережі (рис.2.4). Система складається з трьох основних частин: перша частина – обробка та збереження вхідного зображення, друга частина – витяг ознак на основі Фур'є перетворення і гістограми, класифікація дефектів з використанням нейронних мереж, третя частина відображення результатів класифікації.

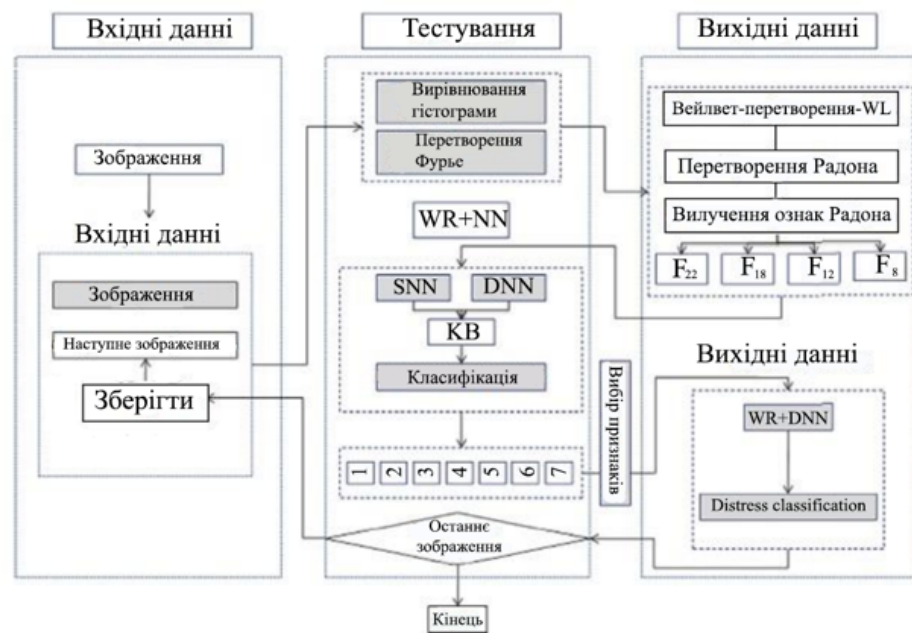


Рисунок 2.4 Система автоматичного обнаруження та класифікація тріщин з використанням динамічних нейронних мереж

## 2.2 Google Maps API

За допомогою Google Maps API в додаток можна додавати карти на основі даних Google. Цей API-інтерфейс автоматично управляє доступом до серверів Google, завантаженням даних, відображенням карт і реакцією на жести, виконувани на картах. Крім того, можна використовувати виклики API, щоб додавати маркери, багатокутники і накладення до основної картки, а також змінювати спосіб відображення певній галузі на карті. Ці об'єкти представляють додаткову інформацію про місця на карті і забезпечують можливості взаємодії користувачів з картою. Щоб використовувати Google Maps API, необхідно зареєструвати проект додатки в Google Developers Console і отримати ключ Google API, який потрібно додати в додаток.

Необхідний тип ключа API - ключ для додатків. Цей ключ надається безкоштовно. Можна використовувати його з будь-яким з додатків, звертаючих до Google Maps API, без будь-яких обмежень за кількістю користувачів.

Основні кроки додавання карти:

1. Отримати ключ, включити ключ API в маніфест додатка. В файлі `AndroidManifest.xml` додати наступний дочірній елемент елемента `<application>`, вставивши його перед закриває тегом `</ application>`: `<meta-data android: name = "com.google.android.geo.API_KEY" android: value = "API_KEY" / >`

2. Додати елемент `<fragment>` в файл макета Activity, щоб визначити об'єкти. У цьому елементі установіть значення `"com.google.gms.maps. MapFragment"` для атрибута `name`. Це автоматично включає `MapFragment` в Activity.

3. Викликати `getMapAsync ()` для фрагмента, щоб зареєструвати обратний виклик. `getMapAsync ()` необхідно викликати з основного потоку, зворотній виклик також буде виконаний в основному потоці. Якщо на пристрої користувача служби Google Play невстановлені, зворотний виклик не буде виконаний до тих пір, поки користувач їх не встановить.

Google Maps API дозволяє вказати, яка частина світу повинна отобра-туватися на карті. Це досягається за допомогою зміни положення камери (на відміну від переміщення карти). Зміни камери не приведуть до якихось змін доданих маркерів, накладень або інших графічних елементів. Модель уявлення карти передбачає направлення камери внізна площину. Положення камери (і, отже, візуалізація карти) вказується за допомогою наступних властивостей: мета (місце розташування з широтою / довготою), масштабування, напрямок і нахил. При зміні камери є можливість застосувати анімацію для її переміщення. Анімація інтерполирует зображення між текущими атрибутами камери і її новими атрибутами.

Також можна управляти анімації. Всі програмні переміщення камери розраховуються щодо відповідності розміру об'єкта `GoogleMap` з урахуванням всіх відступів, доданих до карти. Наприклад, додавання відступу 100 пікселів до лівого краю карти приведе до зміщення центру карти вправо на 50 пікселів. Щоб змінити положення камери потрібно іспользовать `CameraUpdate`. Google Maps API дозволяє

створювати багато різних типів `CameraUpdate` за допомогою `CameraUpdateFactory`. Метод `CameraUpdateFactory.zoomTo (float)` надає об'єкт `CameraUpdate`, який змінює масштаб до заданого значення, зберігаючи всі інші властивості без змін. Існують два допоміжних методи для загальної зміни положення камери. Метод `CameraUpdateFactory.newLatLng (LatLng)` надає `CameraUpdate`, котрий змінює значення широти і довготи камери, зберігаючи всі останні властивості. Метод `CameraUpdateFactory.newLatLngZoom (LatLng, float)` надає об'єкт `CameraUpdate`, який змінює значення широти, довготи і масштабування камери, зберігаючи всі інші властивості. щоб застосувати `CameraUpdate` до камери, можна або переместити камеру моментально, або застосувати до її переміщення плавну анімацію. Для моментального переміщення камери можна викликати `GoogleMap.moveCamera (CameraUpdate)`. Щоб зробити користувацький інтерфейс більш приємним, особливо при невеликих переміщеннях, зміни можна анімувати. Для цього замість виклика `GoogleMap.moveCamera` викликається `GoogleMap.animateCamera`.

## **2.3 Попередня обробка і витяг ознак в системах ВКДП**

### **2.3.1 Попередня обробка зображень в системах ВКДП**

Процес обробки зображень вважається основою системи. система повинна мати можливість працювати з різним дозволом вихідних зображень. Тому процес обробки зображень повинен бути вивчений і оптимізований для того, щоб система виробляла якісні результати, але забезпечувала високу продуктивність системи.

Грунтуючись на виборі методів попередньої обробки, результати виявлення і класифікації системи значно поліпшені, що збільшує швидкість обробки всієї системи. Цей процес включає в себе фільтрацію шуму і отримання плавного зображення зі збереженням країв, виявлення ознак дефектів на зображенні. У даній роботі використаний фільтр Гаусса для видалення шуму. Сутність цього перетворення полягає в реалізації згортки зображення з ядром симетричної форми в вигляді двовимірної функції Гаусса.

Під час реєстрації дефектів зображення неминуче піддаються впливу різних шумом, які в свою чергу впливають на виявлення і витяг ознак дефектів. Крім того,

умови освітлення можуть змінювати різні області зображення, що призводить до їх деградації. Ці проблеми вирішуються за допомогою адаптивної еквалізації гістограм. Алгоритм еквалізації гістограм заснований на простій ідеї і часто використовується при обробці зображень. Мета еквалізації гістограми, як уже зазначалося вище, полягає в збільшенні діапазону яскравості зображення.

У даній роботі використовується морфологічний метод для виявлення дефектів пікселів і видалення невеликих ділянок шуму зображення на зображенні (рис. 2.3). Морфологічний метод аналізу геометричних структур був розроблений для довічних зображень, а потім розширений до напівтонових зображень. Це один з методів, що застосовуються на етапі попередньої обробки. Двома найбільш часто використовуваними операціями є нарощування і ерозія.

Нарощування бінарного зображення  $A$  структурним елементом  $B$  позначається  $A \oplus B$  і задається виразом:  $A \oplus B = \bigcup_{b \in B} A + b$ . В данному вираженні оператор об'єднання можна вважати оператором, що застосовується в околиці пікселів. Структурний елемент  $B$  застосовується до всіх пікселам бінарного зображення.

Ерозія бінарного зображення структурним елементом позначається  $A \ominus B$  і задається виразом:  $A \ominus B = \{z \in A \mid Bz \subseteq A\}$ . При виконанні операції ерозії структурний елемент теж проходить по всім пікселам зображення. Якщо в деякій позиції кожен одиничний піксель структурного елементу співпадає з одиничним пікселем бінарного зображення, то виконується логічне складання центрального пікселя структурного елементу з відповідним пікселем вихідного зображення.

Замикання бінарного зображення структурним елементом позначається  $A \cdot B$  і задається виразом:  $A \cdot B = (A \oplus B) \ominus B$ . операція замикання «Закриває» невеликі внутрішні «дірки» в зображенні, і прибирає поглиблення по краях області.

Розмикання бінарного зображення структурним елементом позначається  $A \circ B$  і задається виразом:  $A \circ B = (A \ominus B) \oplus B$ . Операція ерозії корисна для видалення малих об'єктів і різних шумів, але у цій операції є недолік – всі, хто залишився об'єкти зменшуються в розмірі. цього ефекту можна уникнути, якщо після операції ерозії застосувати операцію нарощування з тим же структурним елементом.

### 2.3.2 Витяг ознак в системах ВКДДП

Як уже було відзначено вище, до вилучення ознак пропонується попередня обробка зображень. Спочатку застосовується фільтрація шуму з використанням фільтра Гаусса і конвертація в напівтонові зображення.

На наступному кроці виконується сегментація зображення для виділення області інтересу.

Використовується поділ зображення на області. Потім відокремлюються пікселі дефектів зображення для виявлення зв'язкових областей. На основі методів математичної морфології розроблено ефективний алгоритм 1 предобробки для виявлення пікселів зі відповідних дефектів і для видалення невеликих областей, які визначені як шум.

Мета методу морфологічної обробки – знайти пов'язані компоненти контуру об'єкта для отримання «ідеального» контуру. В даній роботі використовується наступний алгоритм на основі 8 – зв'язкових форм. Використовується нотація Matlab.

Вихідні дані: Півтонове зображення, структурируючий елемент.

Результат: Контур об'єкта

1. Завантаження півтонування.
2. Перетворення в двійкове зображення з використанням.
3. Нарощування підключеного компонента з структурним елементом «Disk».
4. Замикання області дефектів з структурним елементом («diamond», 1).
5. Заповнення «Filling hole» області дефектів.

Алгоритм 1: Використання оператора морфології для знаходження контуру об'єкта.

а) Витяг ознак в системі ВКДДП.

У даній роботі, розміри дефектів обчислюються в пікселях. Розглянуто ознаки контуру (гістограма ланцюгового коду), моменту зображення. У роботі не передбачені ознаки глибини дефектів. Розглянемо наступні дефекти: Мережа тріщин (рис. 2.5): Взаємозв'язані тріщини, що утворюють серії блоків приблизно прямокутної форми, зазвичай розподілені по великій поверхні дорожнього покриття.

Атрибути є: ширина переважної тріщини (мм), ширина переважної осередки (мм), постраждала площа (м<sup>2</sup>).



Рисунок 2.5 Приклади зображень мереж тріщин

Глибокі тріщини (рис.2.6): Окремо локалізована тріщина в поздовжньому напрямку вздовж дорожнього покриття. Атрибути дефекту поздовжніх тріщин: ширина тріщин (мм), довжина тріщин (м), інтервал тріщин (мм), площа постраждалої площі (м<sup>2</sup>).



Рисунок 2.6 Приклади зображень глибоких тріщин

Вибоїни (рис. 2.7): поглиблення округлої форми різних розмірів в дорожньому покритті. Атрибути є глибина вибоїн (мм) і площа вибоїн (м<sup>2</sup>).



Рисунок 2.7 Приклади вибоїн

З аналізу атрибутів кожного дефекту, були вибрані наступні ознаки: Моменти зображення: моменти можуть бути використані для опису, характеристики, і визначення форми об'єкту в зображенні. Використовуються після етапу сегментації моменти зображення, як правило, беруться з форми об'єкта в зображенні. Описується форма об'єкта і витягується вектор ознак форми для подання форми об'єкта.

Гістограма ланцюгового коду: ГЦК призначена для угруповання дефектів, з використанням методики, аналогічної наглядом людини [98]. Гістограма ланцюгового коду обчислюється з кодовою ланцюжка представленого контуру.

Код ланцюга Фрімена [75] представляє собою компактний спосіб представлення контуру об'єкта. Код ланцюга є впорядкованою послідовністю  $\{ci, i = 1, 2, \dots, n\}$ , де  $ci$  вектор, що з'єднує сусідні пікселі контура. Напрямки  $ci$  кодуються за допомогою цілих значень  $k = 0, 1, \dots, K - 1$  в напрямку проти годинникової стрілки, починаючи від позитивного напрямки  $x - axis$ . Число напрямків  $k$  приймає цілі значення  $2M + 1$ , Де  $M$  є позитивним цілим числом. Ланцюгові коди, де  $K > 8$ , називаються узагальненими ланцюговими кодами [76].

Для автоматичного позначення областей (або відсутності), запропонована система розпізнавання образів з використанням багатовимірного простору ознак. У тозі було обрано такі ознаки гістограм ланцюгового коду, моменти для опису характеристик і визначення форми дефектів в зображенні.

## **2.4 Математичне моделювання та чисельні методи сегментації в системах ВКДП**

Розглянемо найпростішу математичну модель, яка дозволяє сегментувати дефекти. Дана матриця  $A = \{a_{ij}\}$  описує безліч пікселів в позиції

$$\{i, j\} \quad i=1, N, j=1, M,$$

Визначення 2.2.1. Оператор бинаризації (маска) (або сегментації)

$$F: A \rightarrow A \quad b, \text{ где } A \quad b = \begin{cases} a_{ij} \quad b = 0, \text{ без дефекта} \\ a_{ij} \quad b = 1, \text{ если дефект} \end{cases}$$

### **2.4.1 Чисельний метод побудови карти дефектів дорожнього покриття**



Виявлення дефектів дорожнього покриття, де області зображення позначені як містять або не містять дефект і виявлення типу класифікації дефектів дорожнього покриття, де кожному виявленому дефекту дорожнього покриття присвоюються мітки «мережу тріщин», «глибока тріщина» і «Вибойна». Для виявлення дефектів дорожнього покриття на зображенні, значення пікселів розглядаються як: контрастність, яскравість, текстура області дефекту. Отримувані результати у вигляді карти дефектів.

У цьому дослідженні метод розрізу на графах застосований для вирішення задач сегментації зображень, які мають складну структуру на основі аналізу співвідношення компонентної зв'язності між пікселями для побудови карти дефектів. Побудова карти дефектів дорожнього покриття – є маркування пікселів на зображенні значенням (0 – не дефект, 1 дефект) і виконується в 2 етапи (рис. 2.8):

Сегментація зображення алгоритмом мінімального розрізу – максимального потоку. Запропоновано використання алгоритму розрізу на графах [80]. Нехай  $G = (V, E)$  – безліч ребер, видалення яких ділить граф на два ізольованих підграфа, один з яких, зокрема, може бути окремим вузлом. Розріз  $C \in E$  є безліччю ребер, таких, що термінали розділені в орієнтованому графі  $G(C) = \langle V, E - C \rangle$ . Крім того, жодна підмножина не поділяє термінали в  $G(C)$ . «Вартість» розрізу  $C$ , позначається  $|C|$ , і дорівнює сумі його вагових значень

Вихідні дані: Граф  $G = (V, E)$  з ємністю  $c$ , джерело  $s$ , приймач  $t$ .

Результат: Максимальний потік від джерела ( $s$ ) до приймача ( $t$ )

```

1. for  $i := 1$  для всіх ребер  $(u, v)$  do
   $f(u, v) \leftarrow 0$ 
end
2. if  $cf$ 
   $(u, v) > 0$  для всіх ребер  $(u, v) \in$ 
   $cf(p) = \min\{cf(u, v) | (u, v) \in p\};$ 
  else todo 1
end
3. for  $i := 1$  для всіх ребер  $(u, v)$  do
   $f(u, v) \leftarrow f(u, v) + cf(p)$ 
end
```

Алгоритм 2: Опис алгоритму розрізу на графах

На (рис.2.9) результат постобробки сегментації, де  $D$  області дефекту,  $N$  – області без дефекту.

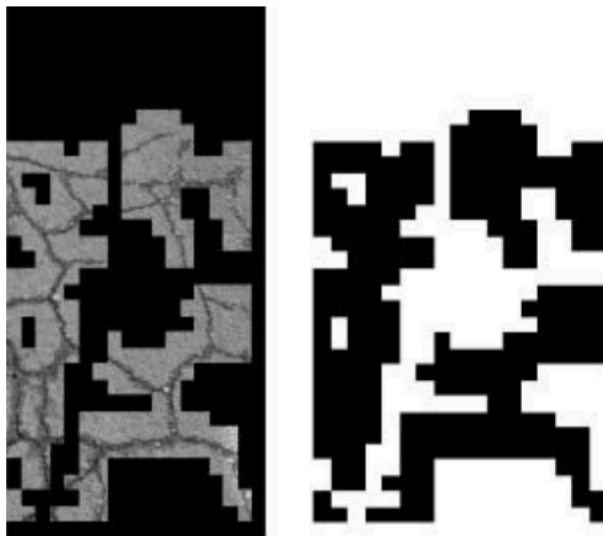


Рисунок 2.9 Використання розрізу на графах для сегментації зображення

Реалізований підхід до сегментації дефектів на основі графів дав можливість використання ефективного методу розв'язання задачі мінімального розрізу – максимального потоку між джерелом і приймачем в орієнтованому графі. Проблема мінімального розрізу в тому, щоб знайти найоптимальніший розріз серед всіх розрізів, що розділяють сегменти. Мінімальні розрізи можуть бути ефективно знайдені за допомогою стандартних комбінаторних алгоритмів з поліноміальної складності, що дозволяє проводити сегментацію навіть складних дефектів за прийнятний час.

Експериментальні результати були отримані за допомогою нового алгоритму мінімального розрізу – максимального потоку, що має кращу швидкість на графах для вирішення багатьох сучасних алгоритмів [34]. Мета полягає в сегментації зображень шляхом побудови графа таким чином, щоб мінімальний розріз задіяв всі ребра, що з'єднують пікселі різних дефектів один з одним.

Алгоритм заміни і розширення для отримання повної карти дефектів дорожнього покриття. Ці сегменти мають попередньо обрану орієнтацію 0, 45, 90 або 135 градусів. Поділ між обома випадками здійснюється за допомогою параметра  $k \in (0,1)$ . За допомогою ітеративного процесу, оптимальні параметри вибираються для

кожного зображення дефекту. Для кожного дефекту, довжина, ширина і орієнтація обчислюються і зберігаються. Наведемо більш докладний опис розробленого ітеративного алгоритму. Заміни значень між пікселями, розширення контуру на основі алгоритму  $\alpha$  – расшірення.

1. Початок з довільної маркування  $f$
2. Встановити змінну  $success := 0$ ;
3. for кожна пара мітки  $D, N \subset L$  do

Найти  $f = \operatorname{argmin} E(f)$  среди  $f'$  в течение одного  $D - N$  замена  $f$ ;

if  $E(f) < E(f')$  then

задавать  $f := f'$  ;

$success := 1$ ; end; end;

4. if  $success := 1$  then

идти к 2;

end

Алгоритм 3: заміни значень між пікселями, розширення контуру На рис. 2.10 результат постобработки сегментації на основі алгоритму заміни і розширення

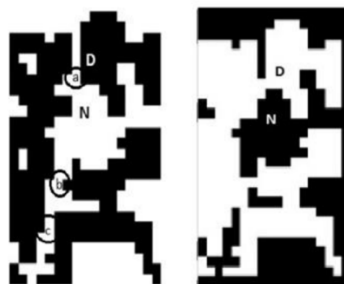


Рисунок 2.10 Результат сегментації зображення на основі алгоритму заміни та розширення

Деякі результати сегментації класів дефектів дорожнього покриття показані на рис.2.11

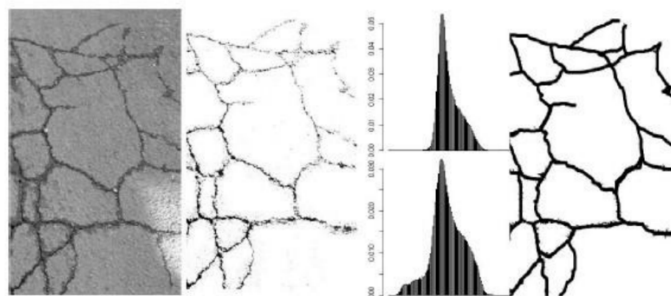


Рисунок 2.11. Результаты сегментации классов дефектов дорожного покрытия

## 2.5 Математичне моделювання та чисельні методи класифікації дефектів

### 2.5.1 Проблема незбалансованих даних

Завдання класифікації з деякою вибіркою називається незбалансованою, якщо дефектів одного або декількох класів в цій вибірці істотно менше, ніж дефектів всіх інших класів. вагове значення класу є важливим параметром настройки для досягнення бажаної продуктивності. У процедурі побудови вирішальних дерев, індекс Джині часто використовується для визначення вагового значення. Розглянемо метрики оцінок, які застосовуються до задачі класифікації [107]: істинно – позитивні – категорії, які ми очікували побачити і отримали на виході, істинноотрицательные – категорії, яких бути на виході не повинно і ці категорії відсутні, помилково – позитивні: категорії, яких бути на виході не має, але аналізатор їх помилково повернув на виході, помилково – негативні – категорії, які ми очікували побачити, але аналізатор їх не визначив. Для алгоритму випадкового лісу завжди існує компроміс між істинноположительними і істинно – негативними і те ж саме застосовується для чутливості (істинно позитивна пропорція, також називається ймовірністю виявлення) і специфічності (істинно негативна пропорція). Набагато більш інформативними метриками в задачах класифікації є точність і повнота

точність  $\frac{ИП+ЛП}{ИП}$  ; повнота  $\frac{ИП+ЛО}{ИП}$  .

У даній роботі використовується крива помилок або ROC – крива – графічна характеристика якості бінарного класифікатора, залежність частки вірних позитивних класифікацій від частки помилкових позитивних класифікацій при варіюванні порога вирішального правила. Перевагою ROC кривої є її інваріантність щодо ставлення ціни помилки I і II роду (Визначення 2.3.1). ROC – крива (горизонтальна вісь – специфічність і вертикальна вісь чутливість) показує залежність ІП від ЛП при варіюванні порога. ROC крива монотонно не убыває. Чим вище лежить крива, тим краще якість класифікації.

**Незбалансовані дані в класифікації дефектів дорожнього покриття.** Навчальний набір складається з 500 зображень, що містять індивідуальні дефекти, що знаходяться на 200 зображень, 300 зображення не містили дефектів (таб. 2.1). У 200

зображень класів дефекту включені 150 зображень (50 мережу тріщин, 50 глибокі тріщини, 50 вибоїни) для тренувального процесу і 50 зображень для процесу тестування. У 300 зображеннях класу без дефектів включені 200 для процесу навчання і 100 зображень для процесу тестування. Дані були отримані за допомогою камери (Canon D500 16 Мп) про стан дорожнього покриття в м. Київ (500 зображень – 300 зображень для навчання і 200 зображень для тестування)). Зображення отримані в звичайному стані денного світла, відстань від камери до поверхні дороги 1 м – 1.2 м.

### 2.5.2 Класифікація дефектів в системі ВКДП

#### Процес створення бази ознак дефектів дорожнього покриття:

Процес створення бази ознак (рис. 2.12) з  $M$  зображень дефектів дорожнього покриття навчальної вибірки.

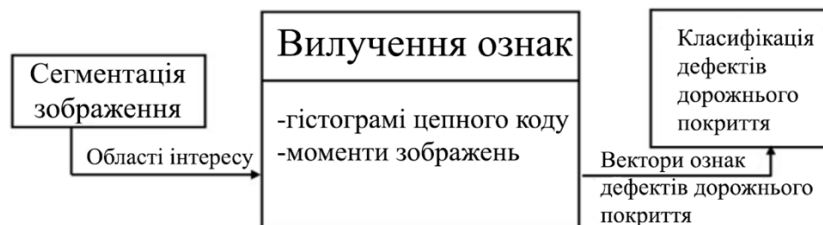


Рисунок 2.12. Блок – схема вилучення ознак та класифікація дефектів

Зображення витягується на області інтересу і витяг ознак складається з наступних кроків:

- Виділення ознак кожного і – ого зображення об'єкта з зображень об'єктів навчальної вибірки на основі аналізу характеристик дефектів.
- Представлення отриманих значень ознак у вигляді вектора
- Витяг ознак:
- Розрахунок гістограми ланцюгового коду являє собою дискретну функцію:  $p(k) = \frac{n}{n_k}$ ,  $k = 0, 1, \dots, K - 1$ , де  $n_k$  є кількістю значень коду ланцюга  $k$  в коді ланцюга, і  $n$  це число ланок в ланцюга коду.
- Розрахунок моментів зображень.

**Процес класифікації дефектів дорожнього покриття:** Нехай  $S = \{X, Y\}$  – безліч  $N$  навчальних вибірок,  $X = \{x_i | i = 1, \dots, 3\}$  – набір векторів ознак дефектів

дорожнього покриття ознак,  $Y = \{y_i | i = 1, \dots, 3\}$  – набір міток класів. Класифікатор – це функція  $H: X \rightarrow Y$ , яка відображає елемент безлічі  $X$  в безліч  $Y$ .

За допомогою алгоритму випадкового лісу будувалася навчальна вибірка, яка задовольнить умові  $H: X \rightarrow Y$ , де  $H$  – класифікатор, а  $Y$  – відповідні мітки. Розглянуто наступні класи дефектів, що задаються безліччю  $Y$ : мережа тріщин, глибокі тріщини і вибоїни. Таким чином, кожне дерево буде підмножиною  $Nt$  з безлічі  $N$ . Кількість вузлів  $Np$ ,  $p \in Np$ , де  $p$  – вузол,  $Np$  – кількість вузлів, які використовуються при навчанні вибірки, ділимо на два класу  $Nl$  і  $Nt$  з урахуванням граничного значення.

Далі, використовуючи максимальний індекс Джині, обчислюємо різницю, яка є критерієм інформативності побудови вирішальних дерев.

Випадковий ліс – це метод МО, заснований на комбінованому навчанні, який є методом створення множинних класифікаторів та синтезує їх результати для отримання кінцевого результату.

Випадковий ліс генерує безліч дерев рішень, а саме алгоритм класифікації і регресії (АКРДР) і використовує метод бегінга. кожне дерево навчається за допомогою бутстрапа, який отриманий з вихідного набору даних. Алгоритм АКРДР побудови дерева рішень будує розбиття простору ознак на непересічні області шляхом його рекурсивного розбиття. Фактично кожному вузлу дерева відповідають деяка область простору ознак і правило, за яким здійснюється її поділ на дві, приписані до дочірнім вершин. Так як завдання побудови оптимального дерева, що мінімізувала помилку на навчальній вибірці при заданому обмеженні на його розміри, є  $N_p$  – важкою, для навчання даної моделі використовується жадібна стратегія максимального зменшення функції, описує неоднорідність даних  $I(D)$ , де  $D$  є область в просторі ознак, відповідна деякому вузлу.

Нехай  $y \in \{1, \dots, K\}$ , де  $K$  – число категорій об'єктів, і область  $D$  містить  $M$  об'єктів з навчальної вибірки, позначимо відносну частоту категорії  $k$  в даному вузлі дерева  $p_k = \frac{1}{M} \sum (y_i = k) I(x_i \in D)$ , де  $I$  – індикаторна функція. Категорія  $k$  з найбільшою відотною частотою відповідає відповіді, пророкувало для всіх об'єктів, що відносяться до даного вузла.

Для створення методів навчання і тестування в випадковому лісі використовувалася техніка «Out – Of – Bag». Він заснований на побудові ансамблю дерев рішень, де кожен елемент будується на вибірці, яка виходить за допомогою бустрепінга – методу дослідження розподілу статистик розподілів ймовірності вихідної навчальної вибірки. Існуючі алгоритми недостатньо ефективні для використання в розробляється системі, тому для вироблення оптимального рішення була введена нечітка експертна компонента. На основі методу «випадковий ліс» розроблений алгоритм класифікації, що дозволяє поліпшити ефективність знаходження класифікованого рішення.

За допомогою методу «випадковий ліс» також можна проводити відбір предикторів і пошук відхилень при аналізі даних. Випадковий ліс разом збегінгом представляють собою ансамбль моделей класифікаторів. У більшості методів формування ансамблів в задачах машинного навчання є спільні риси:

- Конструюються кілька розрізняються прогностичних моделей з адаптованих варіантів елементів навчальної вибірки;
- комбінують передбачення з використанням різних методів, включаючи просте усереднення або зважене голосування.

Навчання моделі випадкового лісу складається з незалежного побудови певної кількості дерев за такою процедурою: генерація випадкової підвибірки з навчальної вибірки (розмір даної підвибірки зазвичай становить 50 – 70% від розміру всієї навчальної вибірки) і побудова дерева рішень по даній підвибірки (причому в кожному новому вузлі дерева змінна для розбиття вибирається не з усіх ознак, а з випадково обраного їх підмножин).

Чисельний метод заснований на алгоритмі випадкового лісу [93] для вирішення проблеми класифікації дефектів дорожнього покриття в системі ВКДДП.

- Крок 1. Для  $i = 1, 2, \dots, B$  ( $B$  – кількість дерев) виконати: – сформувати бустстреп вибірку  $S$  розміру  $l$  по вихідній навчальній вибірці

$$D = \{X_i, y_j\} \quad i = 1 \dots l;$$

– По вибірці  $S$  індукувати неусеченное дерево рішень  $T_i$  з мінімальною кількістю спостережень в термінальних вершинах рівним  $nmin$ , рекурсивно слідуючи наступним алгоритмом:

- з вихідного набору  $n$  ознак випадково вибрати  $p$  ознак;
- з  $p$  ознак вибрати ознака, який забезпечує найкраще розщеплення;
- расщепить вибірку, відповідну оброблюваної вершині на дві підвибірки;
- Крок 2: В результаті виконання кроку 1 отримуємо ансамбль дерев рішень  $\{T_i\} i = 1 B$ ;
- Крок 3: Передбачення нових спостережень здійснювати наступним чином:  
для класифікації: нехай  $w(X) \in \{w_1, w_2, \dots, w_c\}$  – клас, передбачений деревом рішень  $T_i$ , т. Е.  $T_i(X) = w(X)$ ; тоді  $wrf B(X)$  – клас, найбільш часто зустрічається в множині  $\{w(X)\} i = 1 B$ .

В даному розділі описується класифікація, заснована на методі навчання з учителем – алгоритмі випадкового лісу, що застосовує концепцію дерева рішень, виробляючи велику кількість дерев рішень. Спочатку витягується випадкова вибірка даних і визначається ключовий набір ознак, щоб виростити кожне дерево рішень. Ці дерева рішень, тобто їх «Out – Of – Bag» визначають помилку (частота помилок моделі), а потім ряд дерев рішень порівнюється, щоб знайти спільний набір змінних, які виробляють максимально точну модель класифікації.

В якості першого параметра слід явно задати, що клас є саме фактором. Крім кількості дерев приділимо увагу числу ознак (mtry), яке буде використовувати елементарна модель (дерево) для розгалужень. Фактично, це два основних параметри, які є сенс налаштовувати в першу чергу.

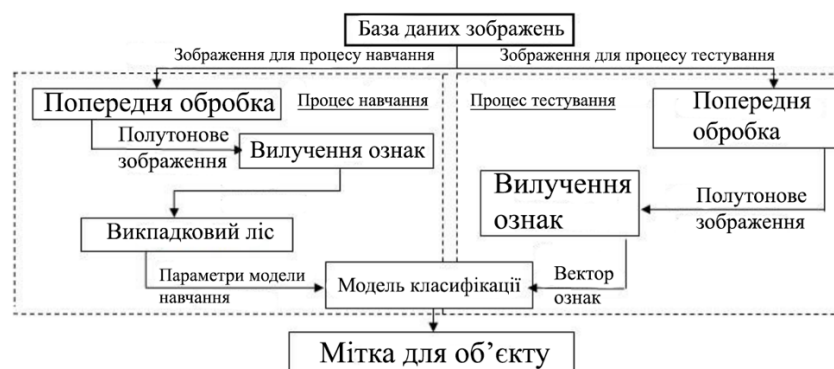


Рисунок 2.13 Блок – схема класифікації дефектів в системі



## 2.6 Застосування методів і алгоритмів ТЗ і МО для виявлення і класифікації об'єктів на зображеннях

У цьому розділі описується процес застосування алгоритмів і методів МО (Рис. 2.13) для вилучення дефектів дорожнього покриття (гістограма ланцюгового коду, локальні шаблони), побудова карти дефектів дорожнього покриття і класифікація даних представлені в розділі 2.7.



Рис.2.14 Блок – схема послідовності виявлення та класифікації дефектів дорожнього покриття

## 2.7 Програмне середовище Swift 5 для розробки мобільної системи

Swift також не відлякає програмістів-початківців. Це перша потужна мова програмування, яку так само легко зрозуміти та розважити, як і мову сценаріїв. Він підтримує так звані ігрові майданчики, які дозволяють розробникам експериментувати з кодом, спостерігаючи за результатами в реальному часі, без компіляції та запуску програми.

Swift включає найкращі з сучасних мов і був розроблений з урахуванням величезного досвіду Apple. Компілятор є синонімом продуктивності, мова оптимізована для програмування без компромісів. Він розроблений таким чином, що ви можете легко створити свій перший додаток "привіт, світ!" і навіть цілу операційну систему. Все це робить Swift незамінним інструментом для розробників та самої Apple.

Переваги Swift :

**Продуктивність.** Swift належить до числа мов високого рівня, які, як відомо, в цілому поступаються по продуктивності низькорівневих мов. В даному випадку, однак, ця тенденція не виправдовується. Програми, написані на Swift, працюють досить швидко. Слово Swift можна перекласти з англійської як "швидкий", яким він

є не тільки з точки зору швидкості розробки, але і з точки зору роботи створюваних на ньому програм. Вони перевершують за швидкістю програми, написані на Objective C більш ніж в 2 рази і майже в 10 разів - програми, написані на скриптових мовах (Python, JavaScript), поступаючись у швидкості поки, хіба що, C ++.

**Безпека.** Впровадження Swift дало м програмами для платформ Apple виграти не тільки в швидкості, але і в безпеці. Пам'ять краще захищена як від несанкціонованого доступу, так і від помилок, пов'язаних з неправильним управлінням апаратними ресурсами комп'ютера. Обробка помилок (винятків) проводиться більш ефективно, що теж знижує число потенційно критичних ситуацій.

**Фінансова привабливість.** Компанія Apple завжди орієнтувалася на заможних споживачів. Її продукція - визнані високотехнологічні предмети престижу, що користуються стійким попитом. Річна виручка від продажу додатків для пристроїв Apple в онлайн-магазині AppStore становить близько 30 мільярдів доларів. Доходи компанії високі, вимоги до якості продукції жорсткі. Це обумовлює необхідність сильно мотивувати створюють програми для пристроїв Apple фахівців. Їхні доходи в середньому традиційно більше, ніж у розробників, які обслуговують інші платформи (Windows, Linux, Android). Майже відразу ж після першого релізу мови Swift попит на розробників, які володіють ним, виріс в 6 разів і залишається стабільно високим. За даними компанії Glassdoor, дохід розробників додатків для iOS перевершує в середньому \$ 100000 в рік.

**Відкритість.** Незважаючи на те, що Swift орієнтований на розробку комерційних додатків, сам він поширюється під відкритою ліцензією, тобто доступний для вільного використання. Навколо Swift сформувалося велике співтовариство програмістів, розвиваюче мову, що надає підтримку починаючим.

### **Висновок до розділу**

У розділі оглянуто математичне моделювання, чисельні методи, засновані на методах ТЗ і МО, комбіновані методи обробки зображень для вирішення завдання виявлення і класифікації дефектів в системах ВКДДП.

Основна увага приділялася завданням обробки даних, попередньої обробки зображень, поліпшення якості вхідних даних з допомогою алгоритмів обробки

зображень. Оцінювалася база даних. У цьому розділі були представлені математичні моделі і чисельні методи, засновані на алгоритмі розрізу на графах і алгоритмі випадкового Марківського поля для сегментування зображень і отримання потрібних областей – областей зображення, які можуть містити дефекти, що підлягають розгляду: дефекти дорожнього покриття.

У дослідженні представлені ознаки дефектів системи (дефекти дорожнього покриття) і метод вилучення ознак.

Також використана математична модель і використовується чисельний метод, заснований на алгоритмі випадкового лісу, для класифікації дефектів.

## РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ТА ЇЇ ПРОГРАМНА РЕАЛІЗАЦІЯ

У цьому розділі дається опис реалізації алгоритмів і методів МН для побудови систем ВКДДП для яких в розділі 2 були розроблені відповідні математичні моделі. Описано структуру програм ВКДДП.

### 3.1 Розробка структурної схеми системи

Структура система складається з окремих модулів. Вона призначена для виконання для забезпечення безперебійної роботи кожного з модулів.

Модульний структури в системі ВКДДП

- Мобільний пристрій;
- GPS датчик для геолокації;
- GSM модуль для сотового зв'язку;
- Модуль додавання зображення;
- Модуль запису геолокації;
- Модуль сегментації зображення;
- Модуль вилучення ознак;
- Модуль класифікації об'єктів;
- Модуль відображення;

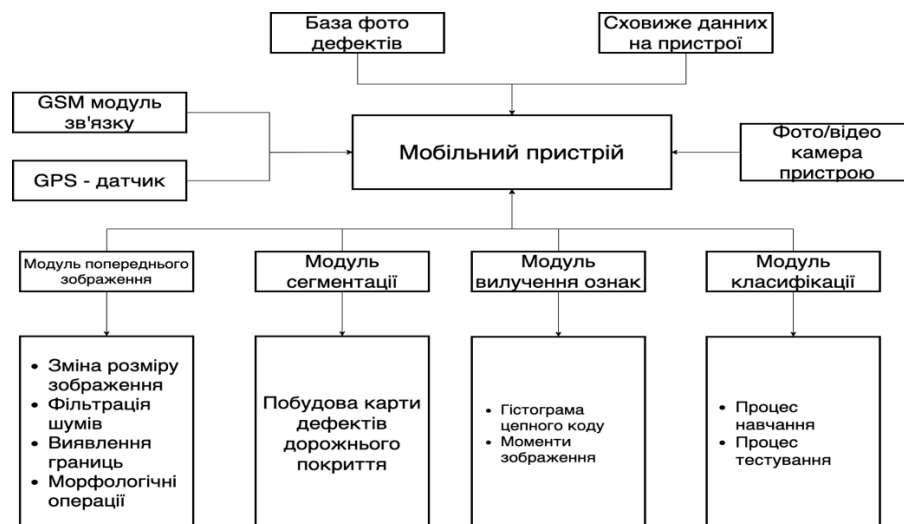


Рисунок 3.1 Структурна схема

### 3.2 Розробка алгоритму роботи системи

Розроблено алгоритм роботи системи, який складається з 4 основних блоків, а саме :

- Додання зображення дефекту у систему за допомогою фото/відео камери;
- Запис геолокації та додання маркіровок на карту;
- Аналіз дефекту дорожнього покриття;
- Вивід звіту про аналіз на робочій екран;

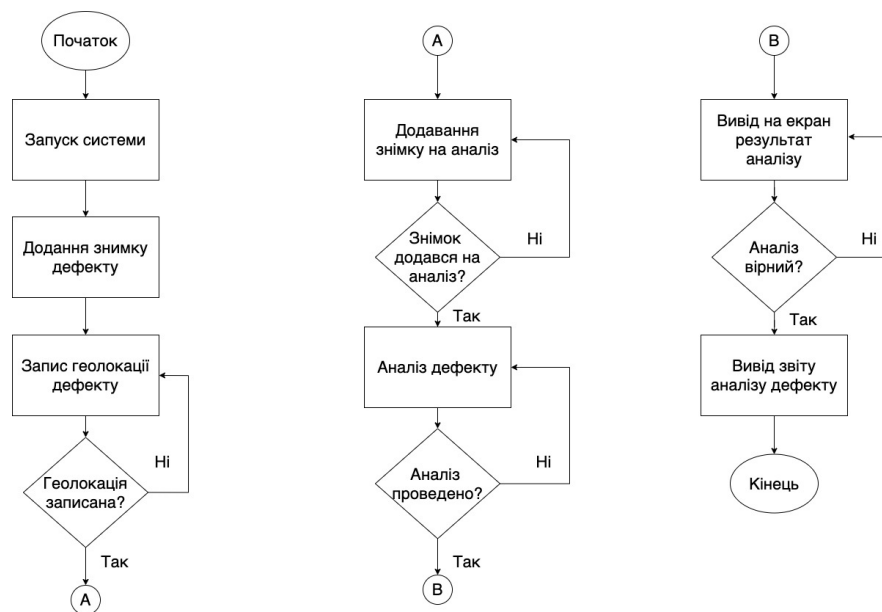


Рисунок 3.2 Алгоритм роботи системи

В результаті проведеного аналізу в розділі запропонована наступна схема для виявлення та класифікації дефектів дорожнього покриття:

Крок 1: попередня обробка зображення

- перетворення в півтонування;
- придушення шуму лінійним фільтром Гаусса;
- еквалізація гістограми;
- використання авторського методу на основі математичної морфології.

Крок 2: сегментація зображення і побудова карти дефектів дорожнього покриття.

- сегментація зображення алгоритмом мінімального розрізу максимального потоку;

- поліпшення сегментованого зображення на основі алгоритму марковських випадкових полів;

- отримання повної карти дефектів дорожнього покриття.

Крок 3: витяг ознак

- гістограма ланцюгового коду;
- моментів зображень;
- створення вектора ознак ВКДДП.

Крок 4: класифікація типів дефектів дорожнього покриття за новими даними

- процес навчання;
- процес тестування;

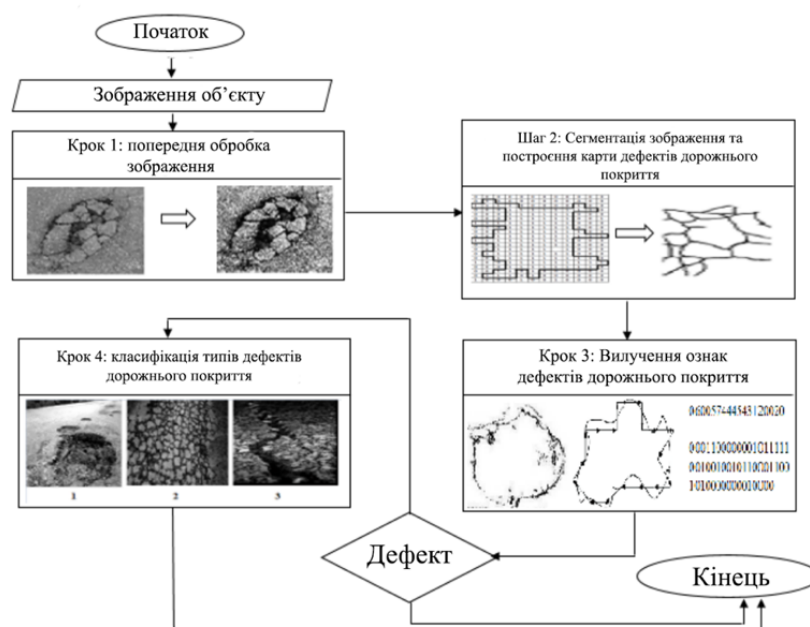


Рис.3.3 Блок – схема виявлення и класифікації дефектів дорожнього покриття в системі

### 3.3 Структура програми виявлення та класифікація дефектів в системах ВКДДП

Програмний комплекс складається з 5 модульних структур ВКДДП (рис. 3.2). Структура системи призначена для забезпечення того, щоб кожен модуль мав працюють функції для виконання однієї і тієї ж задачі і міг розширювати сферу розробки системи науково обґрунтованого методу.

Модульній структури в системі ВКДДП

- Модуль попереднього зображення;
- Зміна розміру зображення;
- Фільтрація шум;
- Виявлення країв;
- Морфологічні операції;
- Еквалізація гістограми;
- Модуль сегментації зображення;
- Модуль вилучення ознак;
- Модуль класифікації об'єктів;
- Процес навчання;
- Процес тестування;
- Модуль аналізу;
- Модуль відображення;
- Побудову карти дефектів;

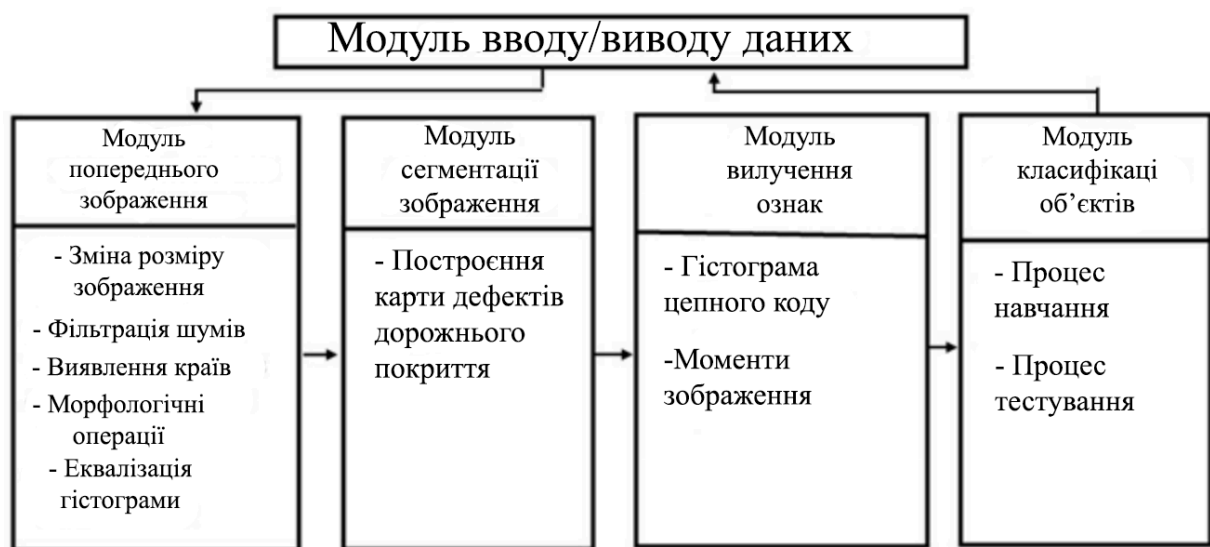


Рисунок 3.4 Схема ПЗ системи знаходження дефектів

### 3.3.1 Модуль введення / виводу даних

Блок введення даних в системі виконує завдання: зберігання зображень, коригування формату і розміру бази даних в стандарті, введення параметрів для алгоритму, відображення результатів системи.

Клас «InOutData» використовується для зберігання зображень і результатів системи. Структура класу наводиться «InOutData» на рис.3.4. Клас «InOutData» наведено в таб.3.1.

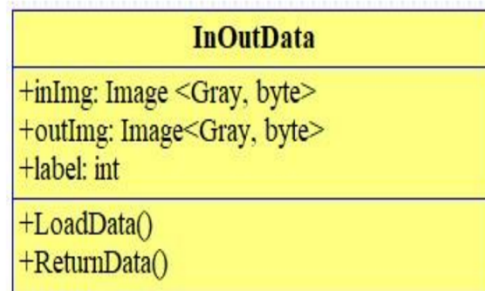


Рисунок 3.5 Структура класу InOutData

Клас констант: призначений для збереження значень констант, таких як посилання до файлів, в яких містяться коефіцієнти вейвлет перетворення Хаара; розміри пікселів кольорового і полутонового зображень; мінімальне і максимальне значення пікселя півтонування. Основні змінні класу констант:

byte GRAY – MIN – VALUE: Мінімальне значення пікселя полутонового зображення.

byte GRAY – MAX – VALUE: Максимальне значення пікселя полутонового зображення.

string WAVELET – HAAR: Посилання до файлу, в якому містяться коефіцієнти для перетворення Хаара.

Таблиця 3.1 Основні змінні і функції класу InOutData

Назва змінної і функції	опис
inImg: Image	Змінна зберігання вхідних зображень.
outImg: Image	Змінна зберігання вихідних зображень.
label	Мітка оброблюваного дефекту.
LoadData()	Функція введення даних в систему
ReturnData()	Функція виведення даних в систему.



**Клас бази даних:** призначений для створення бази зображень і їх відповідних назв, використаних для створення власних відомих об'єктів для навчання. Основні функції класу:

- Database (Image <Gray, Byte> [] images, String [] iLabels): Створення бази зображень.
- List <Image <Gray, byte> »SearchByName (string iName): Знаходження зображень об'єкта по заданому назвою.
- Add (string iName, List <Image <Gray, byte> »images): Додавання зображення об'єкта.
- Delete (string iName, List <int> imagesIndex): Видалення зображення об'єкта.
- Delete (string iName): Видалення об'єкта.

### 3.3.2 Модуль попередньої обробки зображення

Блок попередньої обробки зображення можна розглядати як перший блок в системі для виявлення і класифікації дефектів на зображенні. Отже, дані для сегментації зображення і виявлення дефектів будуть точніше для вилучення ознак блоків. Це істотно впливає на ефективність і надійність всієї системи обробки.

Технологія обробки зображень включає в себе: фільтрацію шуму, еквалізацію гістограм, виявлення меж і морфологічні операції для отримання зображень хорошої якості.

Клас «PreProcessImg» призначений для обробки зображень. цей клас включає такі процеси, як корекцію розміру зображення, фільтрацію шуму, еквалізацію гістограм, морфологічну обробку, виявлення кордонів. Клас «PreProcessImg» представлено на рис.3.3. Основні функції класу «PreProcessImg» наведені в таб.3.2.

LoadData (...): Завантаження даних вибірки в змінних з папки зображень.

Image <Gray, byte> ToGrayImage (Image <Bgr, byte> iImage): Перетворення кольорового зображення в півтонове.

Image <Gray, byte> Resize (Size newSize, Image <Gray, byte> iImage): Зміна розміру зображення iImage до нового розміру.

Image <Gray, byte> FeatureExtract (Image <Gray, byte> iImage): Витяг ознак зображення області об'єкта.

Image <Gray, byte> [] FeatureExtract (Image <Gray, byte> [] iImages): Витяг ознак зображень областей об'єктів.

Recognize (EigenObjects iEigenObjects, Image <Gray, byte> iImage): Розпізнавання зображення області об'єкта з використанням власних відомих об'єктів.

<b>PreprocessImg</b>
+Img: Image <Gray, byte>
+ResizeImg() +FilterGauss() +Morphology() +EqHist() +EdgeCanny()

Рисунок 3.6 Структура класу Preprocessing

Таблиця 3.2 Основні змінні і функції класу PreprocessImg

Назва змінної і функції	Опис
Img: Image	Змінні зображення необхідно обробити.
ResizeImg()	Функція регулювання розміру зображення.
FilterGauss()	Функція фільтруючий шум і гладке зображення.
Morphology()	Функція використовує морфологічні операції
EqHist()	Функція еквалізація гістограм.
EdgeCanny()	Функція виявлення краю

### 3.3.3 Модуль сегментации изображения

Сегментація зображень грає важливу роль в загальній системі. цей процес визначає, які області зображення будуть оброблені і які дефекти будуть ідентифіковані.

- Сегментація зображення алгоритмом мінімального розрізу – максимального потоку, виділення області інтересу.
- Побудова повної карти дефектів методом порівняння і заміни піксельних міток, які отримані на основі алгоритму  $\alpha$  – розширення.

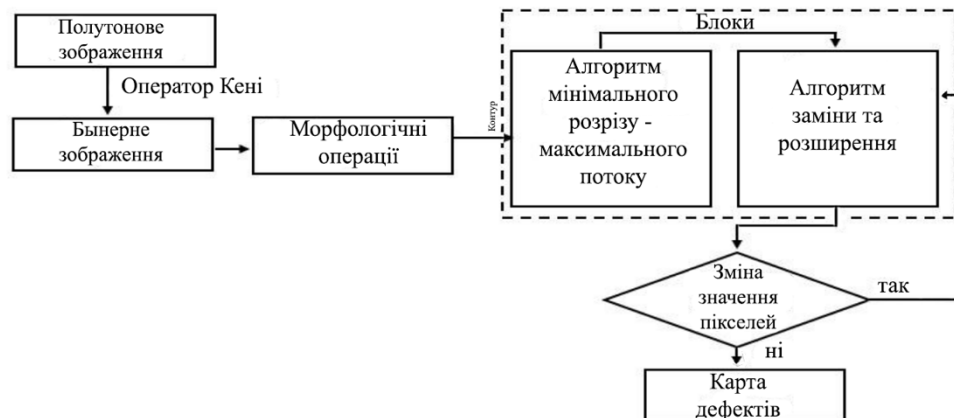


Рисунок 3.7 Блок – схема сегментації зображення на основі розрізу графів

Точність процесу – це фактор, який скорочує час обробки, підвищує надійність і стабільність системи. Цей процес складається з двох етапів: сегментація зображень і побудова карти дефектів дорожнього покриття для систем ВКДДП.

Клас «SegmentImg» призначений для сегментації зображень. цей клас включає в себе такі процеси, як сегментація зображення, побудова карти дефектів дорожнього покриття для систем ВКДДП.

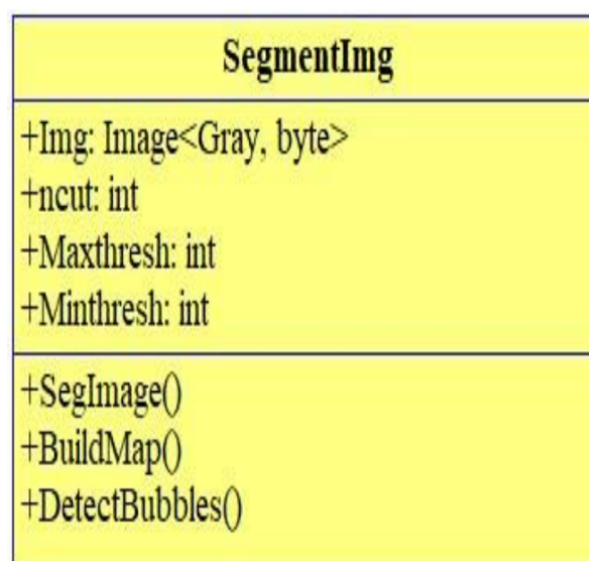


Рисунок 3.8 Структура класу Segmenting

Таблиця 3.3 Основні змінні і функції класу Segmenting

Назва змінної і функції	Опис
Img: Image	Змінна містить зображення для процесу сегментації.
Ncut	Кількість зрізів алгоритму розрізу на графах.
Maxthresh	Максимальне значення порога
Minthresh	Мінімальне значення порога.
SegImage()	Функція сегментації зображення.
Buildmap()	Функція побудова карту дефектів дорожнього покриття для систем ВКДДП.

### 3.4. Верифікація системи виявлення та класифікації дефектів дорожнього покриття

Класифікатор був побудований з використанням параметрів  $n_{tree} = (50, 100)$ ,  $m_{try} = 2$  і  $depth = (2, 5)$ . Випадковий ліс і найшвидшого бустінг над вирішальними деревами (gradient boosting trees, GBT [64]). Для обох алгоритмів, збільшення дерев вимагає більше часу для навчання.

а) База даних дефектів дорожнього покриття;

Ця база містить 550 зображень (350 зображень для навчання і 200 зображень для тестування), розмір зображення  $3264 \times 1836$ . При формуванні розмір зображень і умови зйомки були однаковими.

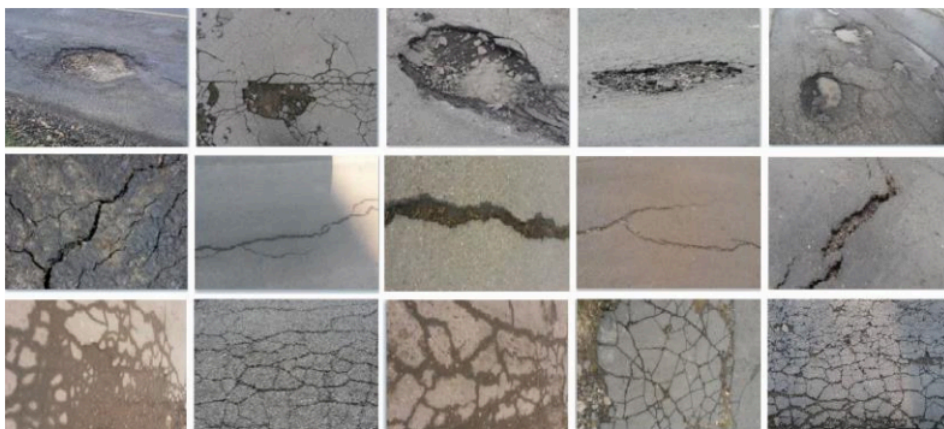


Рисунок 3.9 Приклад зображень з бази даних дефектів дорожнього покриття

Більш розгалужені дерева майже завжди краще, за умови аналогічної продуктивності. Більш розгалужені дерева зменшує відхилення; більшу кількість дерев зменшує дисперсію. Є кілька способів, щоб контролювати наскільки розгалужені дерева (обмеження на максимальну глибину, обмеження кількості вузлів, обмеження кількості дефектів, необхідних для поділу, зупинка розщеплення, якщо поділ йде не в достатній мірі, поліпшення підгонки, і т.д.). У більшості випадків рекомендується підрізати дерева (обмежувати глибину), якщо ми маємо справу з зашумленими даними.

Нарешті, можна використовувати повністю розвинені дерева, щоб обчислити продуктивність коротких дерев, оскільки вони представляють собою «підмножина» повністю розвинених.

Як функціоналу оцінки якості в задачах моделювання і класифікації дефектів обрана ROC – крива.

В цьому випадку для її побудови використовувалася бінарна класифікація послідовно для трьох класів: 1 – вибоїна, 2 – глибока тріщина, 3 – мережа тріщин.

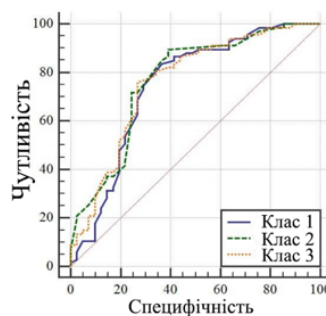


Рисунок 3.10 Графік до першого експерименту

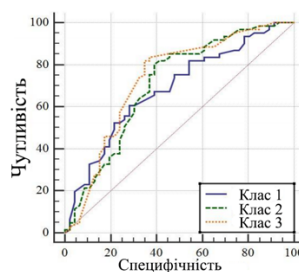


Рисунок 3.11 Графік до другого експерименту

### 3.5 Розробка програмного забезпечення

Було розроблено програмне забезпечення на базі IOS, написане мовою програмування Swift. Було використано середовище розробки XCode для Windows 10.

Для розробки програми було використано бібліотеку Core ML.

Core ML забезпечує надзвичайно швидку продуктивність за допомогою простої інтеграції моделей машинного навчання, що дозволяє створювати програми з інтелектуальними новими функціями, використовуючи лише кілька рядків коду. Легко додайте попередньо вбудовані функції машинного навчання у свої програми, використовуючи API на основі Core ML, або використовуйте Create ML для більшої гнучкості та навчайте власні моделі Core ML прямо на своєму Mac. Ви також можете перетворити моделі з інших навчальних бібліотек за допомогою Core ML Converters або завантажити готові до використання моделі Core ML.

Для початку було побудовано алгоритм роботи програмного забезпечення.

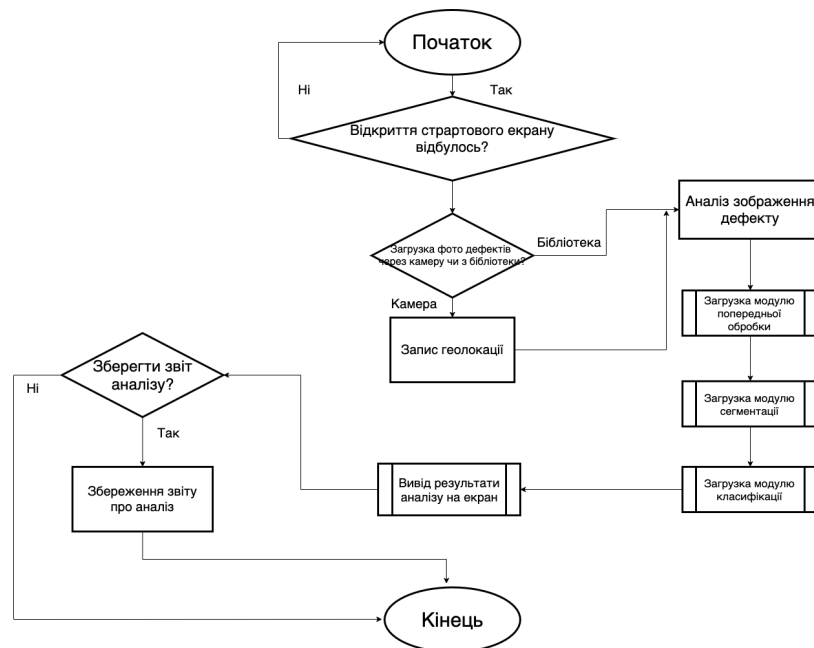


Рисунок 3.12 Алгоритм роботи програми

Спочатку, після відкриття програми відкривається стартовий екран, на якому користувачу дається вибір : завантажити зображення дефекту із бібліотеки чи скористатися камерою.

Якщо користувач вибрав завантажити зображення з бібліотеки, програма переходить до наступного кроку, а якщо через камеру, то спочатку відбувається запис геолокації, де саме було зроблено фото та відбувається маркування на карті.

Після цього відбувається аналіз дефекту дорожнього покриття. В кінці аналізу користувачу надається можливість зберегти звіт. Якщо він вибрав «не зберігати» програми завершується, якщо «зберегти» то зберігається звіт і програма завершується.

### 3.6 Розробка інтерфейсу мобільного додатку

Вихідними даними є результати аналізу зображень, виявлення і класифікації дефектів. Програма дозволяє користувачам використовувати автоматичне виявлення місця розташування дефектів на дорозі, витяг ознак для сортування і показу результатів. інтерфейс користувача:

- «Load image»: завантаження даних (зображення дорожнього покриття);

Завдання набору зображень для тестування, завдання набору зображень для навчання. Для початку роботи з програмою необхідно створити дані, після цього завантажити папку з зображеннями для тестування і навчання. після створення даних відбувається завантаження відповідних файлів. результати розпізнавання об'єктів з'являються на екрані через кілька хвилин в залежності від кількості зображень.

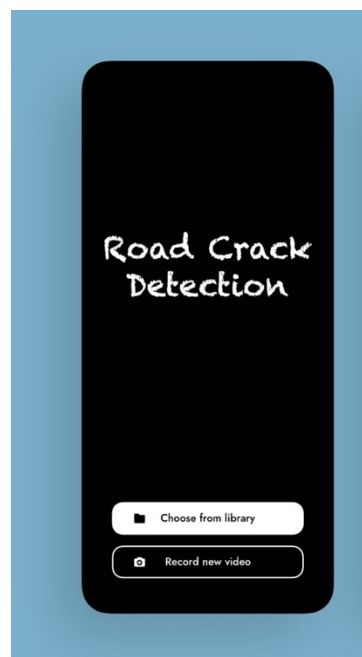


Рисунок 3.13 Скріншот стартового екрану на

Потім для користувача йде варіант вибрати із бібліотеки фотографій чи зробити новий знімок дефекту.

Після того як користувач вибере бібліотеку, то його переведе у галерею, де він вже зможе вибрати потрібне йому зображення.

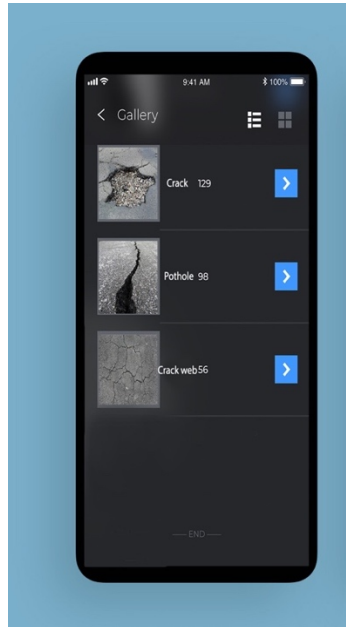


Рисунок 3.14 Вибір дефекту із бібліотеки

Але якщо вибрати «зробити фото» то вас переведе до камери приладу, де можна збільшити/зменшити масштаб, увімкнути сітку або змінити контраст.

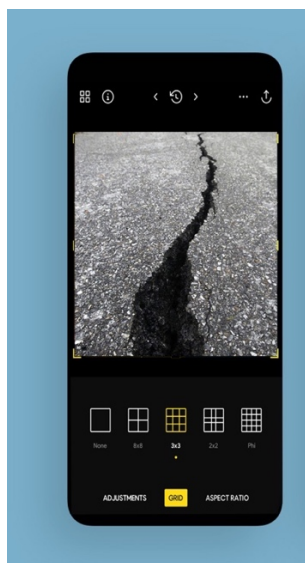


Рисунок 3.15 Зйомка дефектів



Одразу після того, як ви зробили знімок дефекту вас переведе до екрану геолокації, де записуються та макреруються зробленні зображення дефектів. Можна у будь-який час переглянути де і коли було зроблено знімок.

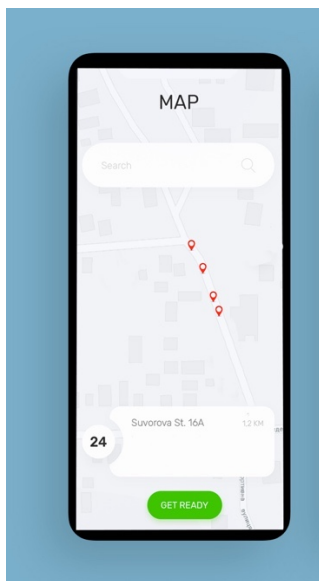


Рисунок 3.16 Запис геолокації після фотографії

Після запису геолокації з'являється екран аналізу, тут проходить класифікації дефекту. На цьому екрані можна змінити фоту та отримати звіт, клікнувши на прямокутник зверху.

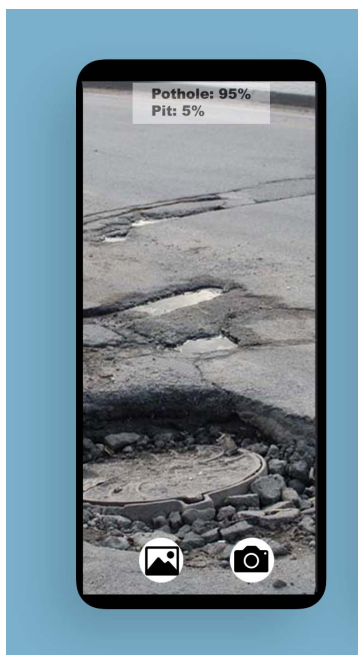


Рисунок 3.17 Аналіз знімку

Також після усіх дій виводить звіт. На ньому можуть бути коментарі щодо дефекту.

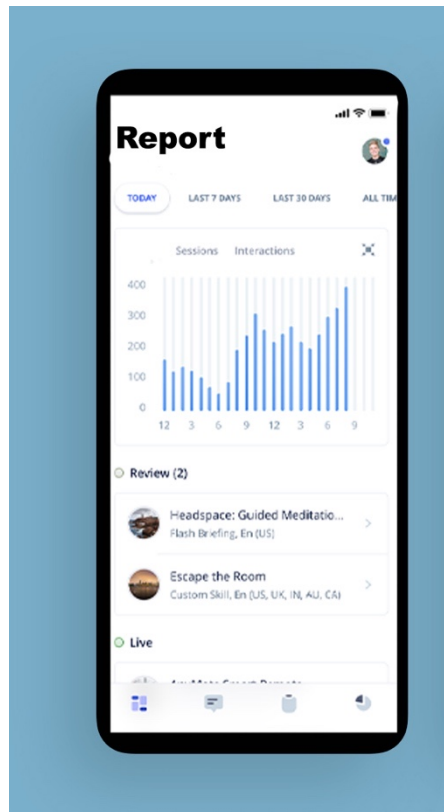


Рисунок 3.18 Звіт про аналіз

### Висновки до розділу

У цьому розділі наведено відомості структури системи для виявлення і класифікації дефектів і класів даних, відповідних блокам: введення і виведення даних, попередня обробка зображень, сегментація зображень, витяг ознак, класифікація дефектів.

Розроблено структуру системи, яка складається з таких модулів як модуль сегментації, модуль додавання зображень та запису геолокації та інші.

Побудовано алгоритм роботи, в якому описано усі основні кроки починаючи від додавання зображення дефекту та закінчуючи виводом звіту аналізу моніторингу дефекта. Також розроблено алгоритми сегментації, виявлення та класифікації дефектів на зображенні дорожнього покриття.

Побудовано інтерфейс мобільного додатку, в якому відображено головні кроки роботи системи.

## РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП – ПРОЕКТУ

### 4.1. Опис ідеї проекту

Таблиця 4.1. Опис ідеї стартап – проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка системи моніторингу якості дорожнього покриття, за допомогою якого технолог зможе на основі самостійно введених налаштувань отримати аналіз та виявити дефекти на дорожньому покритті.	Застосування технологом дорожньої інспекції для швидкого аналізу нових пошкоджень щоб уникнути небезпечних ситуацій на дорозі.	<ul style="list-style-type: none"> <li>– Швидке реагування на команди користувача;</li> <li>– Наявність функцій що упереджають випадкові дії зі сторони користувача;</li> <li>– Наявність функцій збереження та завантаження налаштувань;</li> <li>– Легкість освоєння і застосування;</li> <li>– Баланс між простотою і доступністю функцій і даних;</li> </ul>
	Застосування при проектуванні доріг міжгородського призначення для уникнення дефектів.	

Таблиця 4.2. Опис ідеї стартап – проекту

№	Техніко-економічні характеристики – тики ідеї	Продукція конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	DELMIA	AnyLogic			
1	Швидкість роботи	В	С	Н	Потрібний час на завантаження вікон	Зрозуміла архітектура	Легкість навчання
2	Вимоги системи до	В	С	С	Якісний аналіз	Інтерфейс	Оптимізація
3	Зручність використання	В	С	В	Потрібне розуміння ймовірних дефектів	Наявність функцій збереження та завантаження налаштувань	Легкість освоєння і застосування
4	Точність моделювання	С	В	В	Відсутність функцій зміни розмірів об'єктів	Широкі можливості для позиціонування	Якісний аналіз пошкоджень

## 4.2. Технологічний аудит ідеї проєкту

Таблиця 4.3. Технологічна здійсненність ідеї проєкту

№	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
2	Моделювання виробництва в цілях проєктування	Розробка додатку з упором на широкий функціонал зміни положення та розмірів об'єктів	Використання безкоштовних або платних плагінів з для роботи	У вільному та безкоштовному доступі
3	Моделювання виробництва в цілях розрахунку рентабельності	Розробка додатку з упором на статистичні дані та інфографіки	Використання безкоштовних або платних плагінів з для роботи з статистикою та інфографіками	У вільному та безкоштовному доступі
Обрана технологія реалізації ідеї проєкту: 1				

Висновок: можлива технологічна реалізація продукту, обрана технологія №1.

## 4.3. Аналіз ринкових можливостей запуску стартап – проєкту

Таблиця 4.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн./ум.од	3500
3	Динаміка ринку	Ринок гнучких виробництв та впровадження автоматизації зростає щороку
4	Наявність обмежень для входу	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі або по ринку, %	80%

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження є привабливим.

Таблиця 4.5. Характеристика потенційних клієнтів стартап – проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	имоги споживачів до товару
1	Передбачення виникнення помилок, швидке адітування до виробництва нових деталей або нових умов роботи підприємства	Службовці дорожньої інспекції та звичайні люди	Швидкість роботи з додатком, прагнення отримання результату якнайшвидше	Наявність знань про технічні складові процесу роботи гнучких підприємств
2	Передбачення схеми оптимальної роботи виробництва, передбачення помилок, слабких місць	Спеціалісти що відповідають за проектування	Приділення уваги зручному місцерозташуванню об'єктів виробництва, прагнення передбачити проблемні місця без тестування на реальних об'єктах	Наявність знань про технічні складові процесу роботи гнучких підприємств
3	Розрахунок рентабельності підприємства	Маркетологи гнучких підприємств	Прагнення чітко передбачити ефективність виробництва	

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів які розробляють схожий додаток	Зменшення ціни на програмне забезпечення; Додавання нових, унікальних функцій програмному додатку; Надання ліцензій на обслуговування; Надання своїх спеціалістів;
2	Кошти на розробку та підтримку продукту	Закінчення виділеного бюджету та недостатнє фінансування	Участь в ІТ – конкурсах задля привертання уваги та нових інвесторів до програмного продукту; Мотивація роботи на перспективу; Розробка додатку частково, по ітераціям, щоб покроково виводити на ринок, пригортати до себе увагу потенційних інвесторів та отримувати відгуки від користувачів;
3	Вихід аналогу	Вихід аналогу програмного додатку	Випуск програми у коротші строки з неповною функціональністю для зацікавлення цільової аудиторії; Проведення рекламної компанії;

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нових функціональних можливостей; Випуск нового програмного забезпечення на ринок;
2	Вихід аналогу	Надати програмне забезпечення з новими характеристиками та можливостями що відсутні у програмі конкурентів	Аналіз ринку та цільової аудиторії щоб задовільнити їх потреби та надати функціональність у найкоротші строки за ціну, яка є дешевшою ніж у конкурентів.
3	Зворотній зв'язок від користувачів	Можливість отримання необхідної інформації для виправлення багів та вдосконалення продукту	Отримання вхідних даних допрацювання програмного забезпечення задля задовільнення потреб кінцевих користувачів.
4	Грошова винагорода за рекламу	При достатньому попиту на програмний додаток можлива комерціалізація продукту на основі реклами	Введення реклами; Точкова комерціалізація продукту; Влиття додаткових коштів у проект щоб забезпечити його подальший розвиток.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Товар від кожної компанії на ринку, являється недосконалим замінником товару, реалізованого іншими фірмами; На ринку є умови для входу та виходу; Ціна корелює між суперниками;	Розробка продукту з характеристиками, які покривають сфери застосування що не покривають інші додатки; Кореляція цін у відповідності до інших додатків;
2	Рівень конкурентної боротьби: світовий	Всі продукти замінники розроблялись інтернаціональними командами з різних куточків світу, продукти не належать до певної держави, а належать команді розробників	Вихід на ринок збуту продукту з клієнто – необхідною функціональністю; Налагодження маркетингу на основних інтернет – ресурсах задля охоплення великої кількості потенційних користувачів; Надання бета – версій продукту.

Закінчення таблиці 4.8

3	Конкуренція за видами товарів: товарно – видова	Дана конкуренція – конкуренція між товарами одного виду.	Впровадження функціональності яка відсутня у інших додатків; Спрощення інтерфейсів; Надання підтримки.
4	Характер конкурентних переваг: цінова та не цінова	Цінові переваги – точкова комерціалізація; Не цінова – надання функціональності, що відсутня у товарах – замінниках.	Надання платних ліцензій лише на критично важливу функціональність для клієнта з певним строком підтримки, що зазначена у відповідній ліцензії; Впровадження унікальної функціональності.
6	За інтенсивністю: марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів – замінників	Впровадження власної назви та власного знаку.

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари – замінники
	Anylogic, DELMIA	Відсутні	The AnyLogic Company, Dassault Systems	Онлайн – магазини що продають додатки	Програмні додатки що підтримують 3D – моделювання та написання алгоритмів
Висновки	Розширюють свою діяльність, концентруються на різних напрямках програмних продуктів	Відсутні	Постачальники контролюють розробку програмних продуктів, відповідають за рекламу.	Клієнти диктують ринкові умови, надають розробникам відгуки та нові цілі для покращення додатків	Можливість покращення програмного продукту

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів – аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Швидке реагування на команди користувача.	Швидкість та зручність роботи з програмою – потужна перевага будь – якого конкурентноспроможного додатку.
2	Наявність функцій що упереджають випадкові дії зі сторони користувача.	Перед кожною важливою дією, результат якої виправити буде неможливо або це займе більше часу, додаток від користувача буде очікувати додаткового підтвердження
3	Наявність функцій що дозволяють проводити аналіз за допомогою нейронних мереж.	Основна задача додатку – відобразити якісний аналіз.
4	Доступність функцій маніпулювання об'єктами в середовищі додатка.	Користувач може самостійно розташовувати цехи в середині середовища, змінювати їх положення та кут повороту.
5	Наявність функцій збереження та завантаження налаштувань.	Щоб пришвидшити роботу з програмою та випадково не втратити введені налаштування в додатку доступні функції збереження вікон інтерфейса та інших налаштувань.
6	Легкість освоєння і застосування.	Інтуїтивно зрозумілий інтерфейс, приємні кольори та проста архітектура програми дозволить будь – якому користувачу з легкістю освоїти додаток.
7	Однаковість методів роботи з системою, однаковість стилів та кольорів інтерфейсу користувача.	Простота в освоєнні програми, частини інтерфейсу що запам'ятовуються інтуїтивно – дуже важлива частина дружнього інтерфейсу користувача.
8	Оптимізація.	Швидкість відгуку програми, швидке виведення результатів та миттєве відображення якісної 3Dграфіки привабить користувача.



Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін додатку

№	Фактор конкурентоспроможності	Бали 1 20	Рейтинг товарів – конкурентів у порівнянні з запропонованим						
			- 3	- 2	- 1	0	+1	+2	+3
1	Швидке реагування на команди користувача.	3			AL	DM			
2	Наявність функцій що упереджають випадкові дії зі сторони користувача.	5				AL	DM		
3	Доступність функцій маніпулювання об'єктами в середовищі додатка.	4		DM	AL				
4	Наявність функцій збереження та завантаження налаштувань.	0		AL				DM	
5	Легкість освоєння і застосування.	9	DM	AL					
6	Однаковість методів роботи з системою, однаковість стилів та кольорів інтерфейсу користувача.	0				AL	DM		
7	Оптимізація.	8	DM			AL			

Таблиця 4.12. SWOT аналіз стартап – проєкту

<p>Сильні сторони (S):</p> <p>Швидке реагування на команди користувача</p> <p>Функції що дозволяють моделювати виробництво у 3D – середовищі з якісною графічною системою</p> <p>Легкість освоєння і застосування</p>	<p>Слабкі сторони (W):</p> <p>Вузьконаправленність</p> <p>Можливе виникнення системних помилок</p> <p>Не може працювати на слабких системах</p>
<p>Можливості (O):</p> <p>Приватні контракти</p> <p>Державні контракти</p> <p>Постійна підтримка та розширення функціональності додатку</p>	<p>Загрози (T):</p> <p>Складність передбачення роботи алгоритмів при непередбачуваних ситуаціях</p> <p>Потужні конкуренти</p>

Таблиця 4.13. Альтернативи ринкового впровадження стартап – проєкту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс – наявний	1 – 3 місяці

2	Реклама	Залучення власних коштів для реклами товару	1 місяць
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс – наявний	2 – 3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні	1 – 2 місяці

#### 4.4. Розроблення ринкової стратегії проєкту

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Технологи на гнучких підприємствах	Присутня	Високий	Середня	Легка
2	Спеціалісти що відповідають за проектування структури гнучких підприємств	Присутня	Високий	Висока	Середня
3	Маркетологи гнучких підприємств	Присутня	Низький	Висока	Середня
Які цільові групи обрано: 1, 2, 3					

Отже, відповідно до проведеного аналізу, основною цільовою аудиторією для розповсюдження додатку є працівники гнучких підприємств, які відповідають за безперервну обробку жеталей, проектування підприємств та маркетологи цих підприємств. Обрано стратегію таргетингового маркетингу, оскільки програмне забезпечення підходить тільки для вузьконаправлених підприємств та працівників. Клієнтам буде надаватися пробна версія продукту, повну версію програми можна буде придбати з можливістю розширення функціональності за домовленістю.

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
--------------------------------------	---------------------------	--	---------------------------

Надання унікальної вузьконаправленої функціональності яка відсутня у конкурентів, надання свого мпеціаліста	Участь у конкурсах, конференціях, проведення рекламної компанії.	Можливість швидко додати потрібні для користувачів функції на основі їх відгуків; Якісна та оптимізована аналітика; Відмітні характеристики товару;	Стратегія диференціації
---	--	---	-------------------------

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є схожі додатки, але вони не мають повного функціоналу	Так, ціль компанії знайти нових клієнтів та забрати вже існуючих у конкурентів задля збільшення продажів	Основна ціль – це розробка нового унікального функціоналу а також запозичення важливих та потрібних функцій у додатківконкурентів	Стратегія лідера

Таблиця 4.17. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап – проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Ефективність	Оновлення та розширення	Перспективи розширення функціоналу та розвиток проекту	Швидке освоєння
2	Точність моделювання	Оновлення та розширення	Легке освоєння та доступність функцій збереження налаштувань	Якісна робота алгоритмів
3	Наглядна симуляція	Заняття місця конкурентів	Швидке реагування на дії користувача, оптимізація коду та графіки	Якісна аналіз

Отже, можна зробити висновок, що стартап – компанія обирає як базову стратегію розвитку – стратегію оновлення та розширення, як базову стратегію конкурентної поведінки – стратегію заняття місця конкурентів.

#### 4.5. Розроблення маркетингової програми стартап – проекту

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	На виробництві	Швидкість роботи з додатком, швидке отримання потрібного результату	Додаток направлений на моделювання вузьконаправлених підприємств, тому в ньому заздалегідь встановлені певні налаштування та функції для зручності
2	При проектуванні	Попереднє передбачення недоліків системи, слабких та неоптимальних місць, визначення об'ємів виробництва	Робота з додатком направлена на роботу з вузьким колом можливих виробництв, тому змодельовавши систему проектувальник одразу може побачити більшість недоліків

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Система моделювання гнучкого виробництва		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Швидкість роботи	Нм	Тл
	Легкість освоєння	Нм	Тл
	Оптимізація	Нм	Тх
	Якість 3D – графіки	Нм	Е
	Зручність	Нм	Ор
	Точність моделювання	Нм	Тх
	Дружній інтерфейс	Нм	Е
	Додаток поставляється у вигляді .exe – файлу з інсталятором		
	Марка: RealPlan		
3. Товар із підкріпленням	До продажу: наявні промокоди на придбання декількох ліцензій, знижки на покупку товару, повна документація.		
	Після продажу: додаткова підтримка з боку розробників, надання своїх спеціалістів для допомоги в налаштуванні		
За рахунок чого потенційний товар буде захищено від копіювання: патент, захист інтелектуальної власності.			

Таблиця 4.21. Визначення меж встановлення ціни

Рівень цін на товари – замітники	Рівень цін на товарианалоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
500 – 2000\$	100 – 2000\$	1000 – 1000000\$/місяць	60 – 200\$

Таблиця 4.22. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Клієнти зможуть придбати додаток на основі ліцензії	Забезпечити можливість придбати додаток кожному бажаючому	Розробник – посередник – клієнт	Роздрібна

Таблиця 4.23. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Технологи	Інтернетфоруми, соціальні мережі, конференції, приватні зустрічі	Швидкість моделювання	Демонстрація широкого функціоналу додатку	Рекламне звернення повинно бути націлено на інтереси потенційних клієнтів, статистику, інфографіки та якість
2	Проектувальники	Інтернетфоруми, соціальні мережі.	Точність моделювання	Демонстрація якості.	інтереси потенційних клієнтів, статистику, інфографіки та якість
3	Маркетологи	Інтернетфоруми, соціальні мережі, конференції, приватні зустрічі	Виведення статистики, економія	Демонстрація статистичних таблиць в результаті роботи з додатком	інтереси потенційних клієнтів, статистику, інфографіки та якість

Як результат було створено маркетингову програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару.

## **Висновки по розділу**

В розділі детально описано маркетинговий аналіз стартап – проекту. А саме: описано ідею проекту, розглянуто технологічний аудит ідеї проекту, технологічну здійсненність ідеї проекту, проведено аналіз ринкових можливостей – дано попередню характеристику потенційного ринку та характеристику потенційних клієнтів проекту. Описано фактори загроз, фактори можливостей, зроблено ступеневий аналіз конкуренції на ринку, аналіз конкуренції в галузі за М. Портером. Обґрунтовано фактори конкурентоспроможності, зроблено порівняльний аналіз сильних та слабких сторін додатку. Проведено SWOT – аналіз проекту, визначено альтернативи ринкового впровадження. Розроблено ринкову стратегію проекту: обрано цільові групи потенційних споживачів, визначено базову стратегію розвитку, базову стратегію конкурентної поведінки, визначено стратегію позиціонування. Розроблено маркетингову програму стартап – проекту: визначено ключові переваги концепції потенційного товару, описано три рівні моделі товару, визначено межі встановлення ціни та зформовано систему збуту. Описано концепцію маркетингових комунікацій.

## ВИСНОВКИ

В рамках магістерської дисертації вивчено предметну область, визначено предмет та об'єкт дослідження, розглянуто актуальність та описано новизну продукту. Проаналізовано існуючі рішення – системи моніторингу якості дорожнього покриття, описано їх функціонал та недоліки. На основі цієї інформації було поставлено задачу дисертації та визначено вимоги до програмного забезпечення, що буде розроблено з врахуванням усіх недоліків вже існуючих рішень.

В першому розділі магістерської дисертації були розглянуті існуючі аналоги системи на ринку. В результаті цього було виявлено недоліки цих систем та поставлена задача на розробку системи, яка буде вигравати на ринку над ними за допомогою вилучення цих недоліків. Також розглянуто види картографічних сервісів, які можна використовувати для геолокації зображення.

У другому розділі було проаналізовано типи машинного навчання та виявлено підходящих тип. Було розглянуто математичну модель системи. Було підібрано модуль геолокації, а саме Google Maps API, яких буде використовуватися для макрування зображень дефектів. Розглянуто методи класифікації та сегментації зображень.

В рамках цієї роботи було використано фото – матеріали, які були створені власноруч, задля проведення тестувань та навчання нейронної мережі.

У третьому розділі було побудовано структуру системи, яка складається з основних модулів додавання та аналізу дефектів. Розроблений основний алгоритм роботи системи, також алгоритм сегментації зображення та виявлення і класифікації зображення дефекту дорожнього покриття.

В ключі розробки програмного забезпечення був наданий алгоритм та представлена схема цього алгоритму.

Було побудовано інтерфейс для мобільних пристроїв на операційній системі IOS.

Описано маркетинговий аналіз стартап – проекту – ідею проекту, розглянуто технологічний аудит ідеї проекту, технологічну здійсненність ідеї проекту, проведено аналіз ринкових можливостей – дано попередню характеристику потенційного ринку

та характеристику потенційних клієнтів проекту. Описано фактори загроз, фактори можливостей, зроблено ступеневий аналіз конкуренції на ринку, проведено аналіз конкуренції в галузі за М. Портером. Обґрунтовано фактори конкурентоспроможності, зроблено порівняльний аналіз сильних та слабких сторін додатку. Проведено SWOT – аналіз проекту, визначено альтернативи ринкового впровадження. Розроблено ринкову стратегію проекту: визначено базову стратегію розвитку, проаналізовано цільові групи потенційних споживачів, базову стратегію конкурентної поведінки, визначено стратегію позиціонування. Розроблено маркетингову програму стартап – проекту: визначено ключові переваги концепції потенційного товару, описано три рівні моделі товару, визначено межі встановлення ціни та зформовано систему збуту. Описано концепцію маркетингових комунікацій.



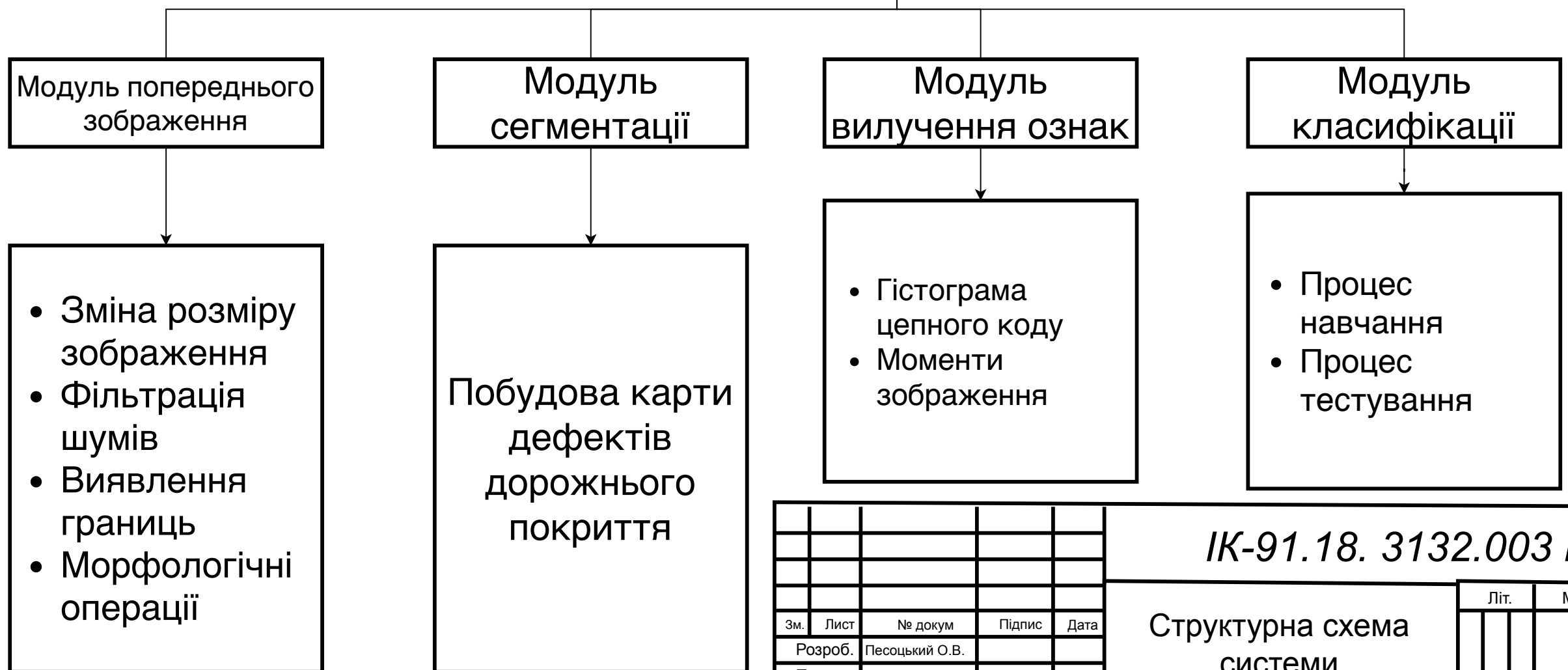
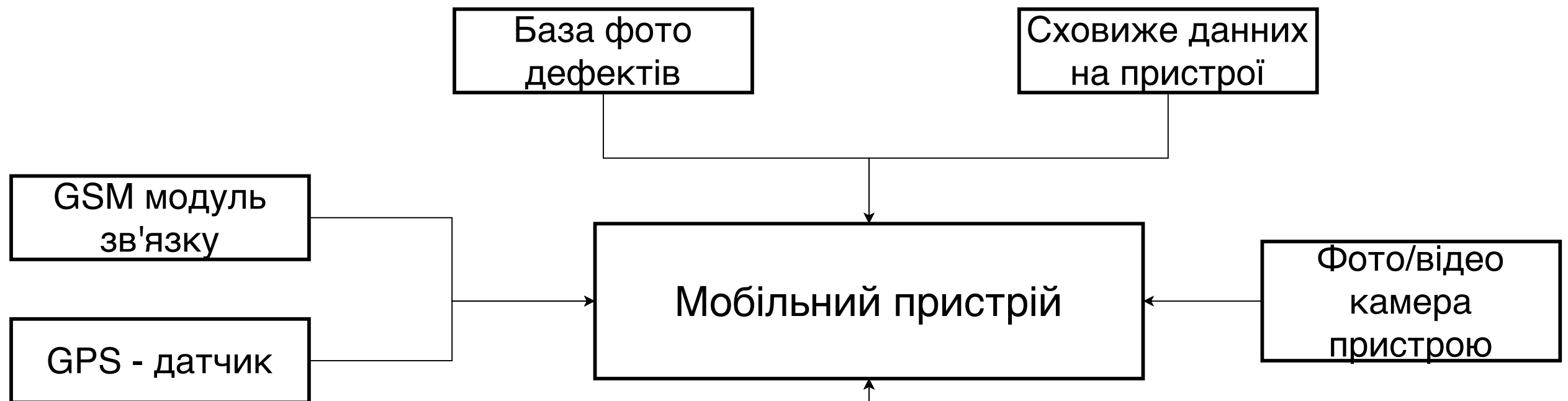
## ПЕРЕЛІК ПОСИЛАНЬ

1. Абрамов, Н. С. «Розпізнавання на основі інваріантних моментів» / Н. С. Абрамов, В. М. Хачумов // «Інформаційні технології»;
2. Белоусов, А. А. Застосування генетичних алгоритмів вейвлет перетворень для підвищення якості зображення;
3. Голубєв, М. Н. «Використання інформації про колір в алгоритмі виділення осіб на базі бустінга» / М. М. Голубєв // «Нейроінформатика»;
4. Гонсалес, Р. «Цифрова обробка зображень»;
5. Кузьмін, Д. М. «Технологія і методи інтелектуального моніторингу автотранспортних потоків і стану автомобільних доріг»;
6. Нгуєн, Т. Х. «Про розпізнаванні і класифікації дефектів дорожнього покриття на основі зображень»;
7. Barker, S. A. «Unsupervised image segmentation using Markov random fields Models»;
8. Bray, J. «A neural network based technique for auto – matic classification of road Cracks»;
9. Chambon, S. «Introduction of a wavelet transform based on 2D matched filter in a Markov randomfield for fine structure extraction: Application on road crack detection»;
10. Кузьмін, Д. М. Технологія і методи інтелектуального моніторингу автотранспортних потоків та якості автомобільних доріг. [Text] / Д. М. Кузьмін. — 2008. — С. 191. 9.
11. Лемперт, А. А. Алгоритм побудови оптимальних поверхностей рівними кругами неопуклих багатокутників з неевклідовою метрикою [Текст] / А.

## ДОДАТКИ

## ДОДАТОК А.

### Структурна схема системи

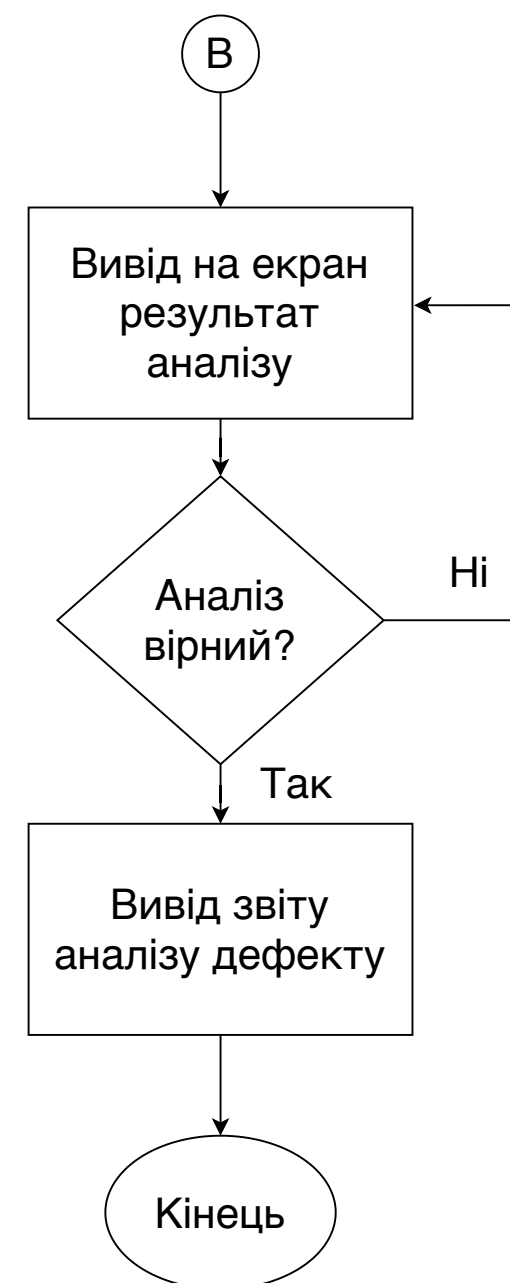
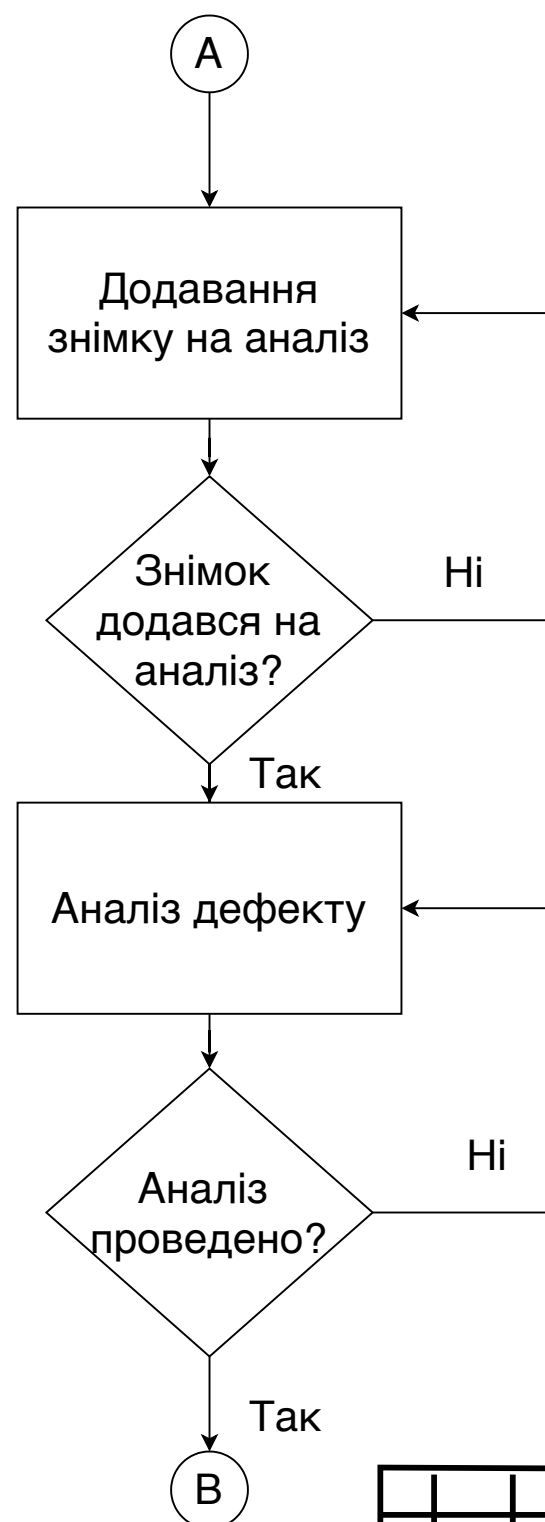
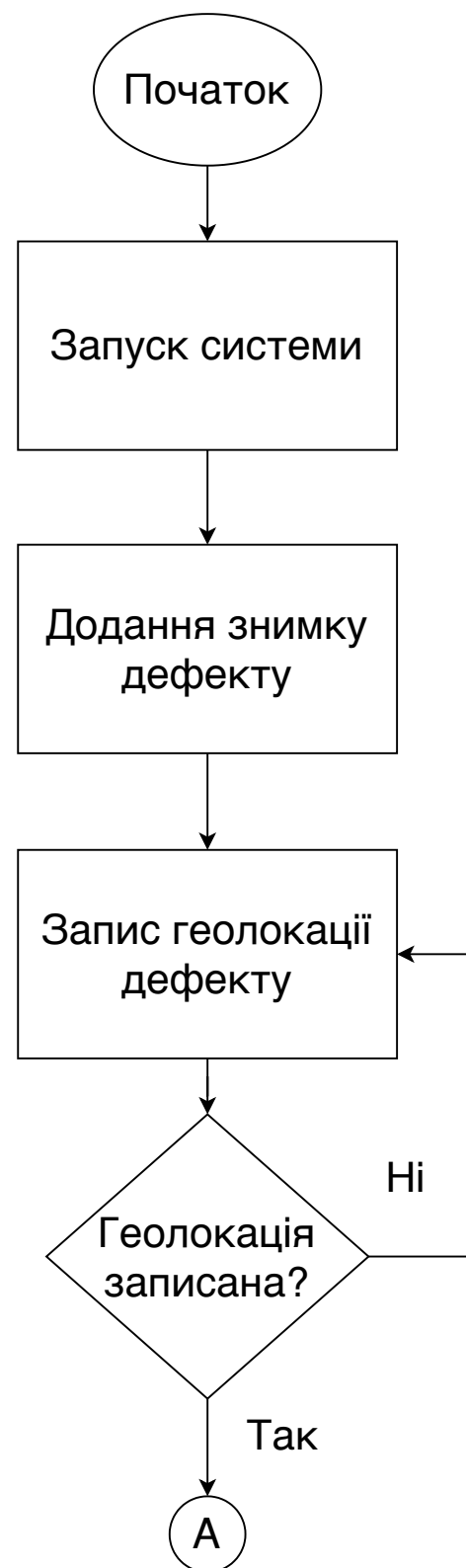


					ІК-91.18. 3132.003 ВЗ									
					Структурна схема системи					Літ.		Маса	Мірило	
Зм.	Лист	№ докум	Підпис	Дата						Лист		1	Листів	1
Розроб.		Песоцький О.В.												
Перев.		Мелкумян К.Ю.												
Н.контр		Пасько В.П			Кафедра Технічної кібернетики					Група ІК-91мп				
Затв.		Пархомей І.Р.												

Інв. № ориг.	Взам. інв. №	Інв. № дубл.	Підпис і дата

## ДОДАТОК Б

### Алгоритм роботи системи

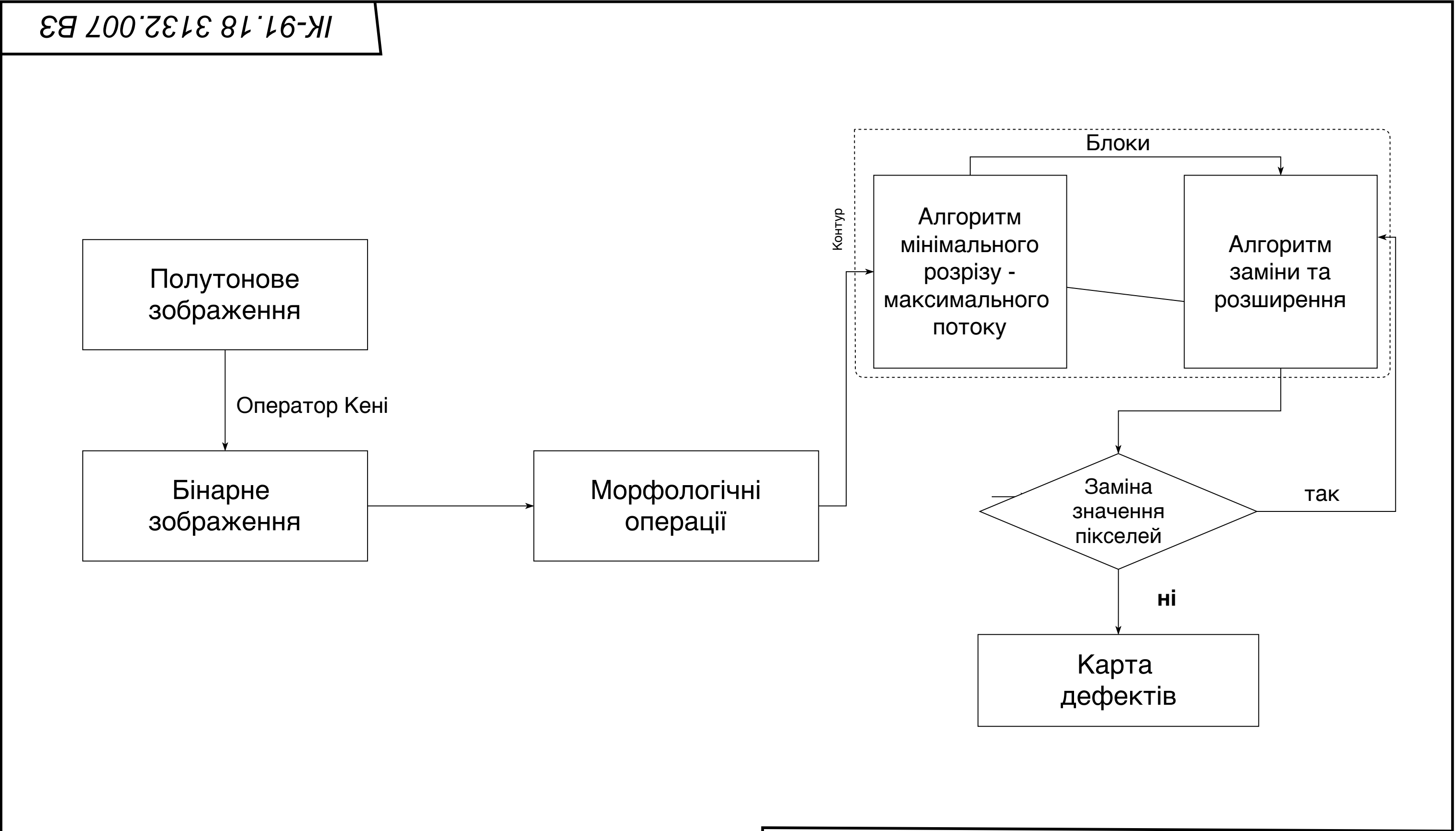


					ІК-91.18. 3132.003 ВЗ										
					Структурна схема системи					Літ.		Маса		Мірило	
Зм.	Лист	№ докум		Підпис						Дата					
Розроб.	Скрипка Д.С.														
Перев.	Польшакова О.М														
										Лист	1	Листів	1		
					Кафедра Технічної кібернетики					Група ІК-91мп					
Н.контр	Пасько В.П														
Затв.	Пархомей І.Р														

## ДОДАТОК В

Блок – схема сегментації зображення

	Підпис і дата		Інв. № дубл.		Взам. інв. №		Підпис і дата		Інв. № ориг.

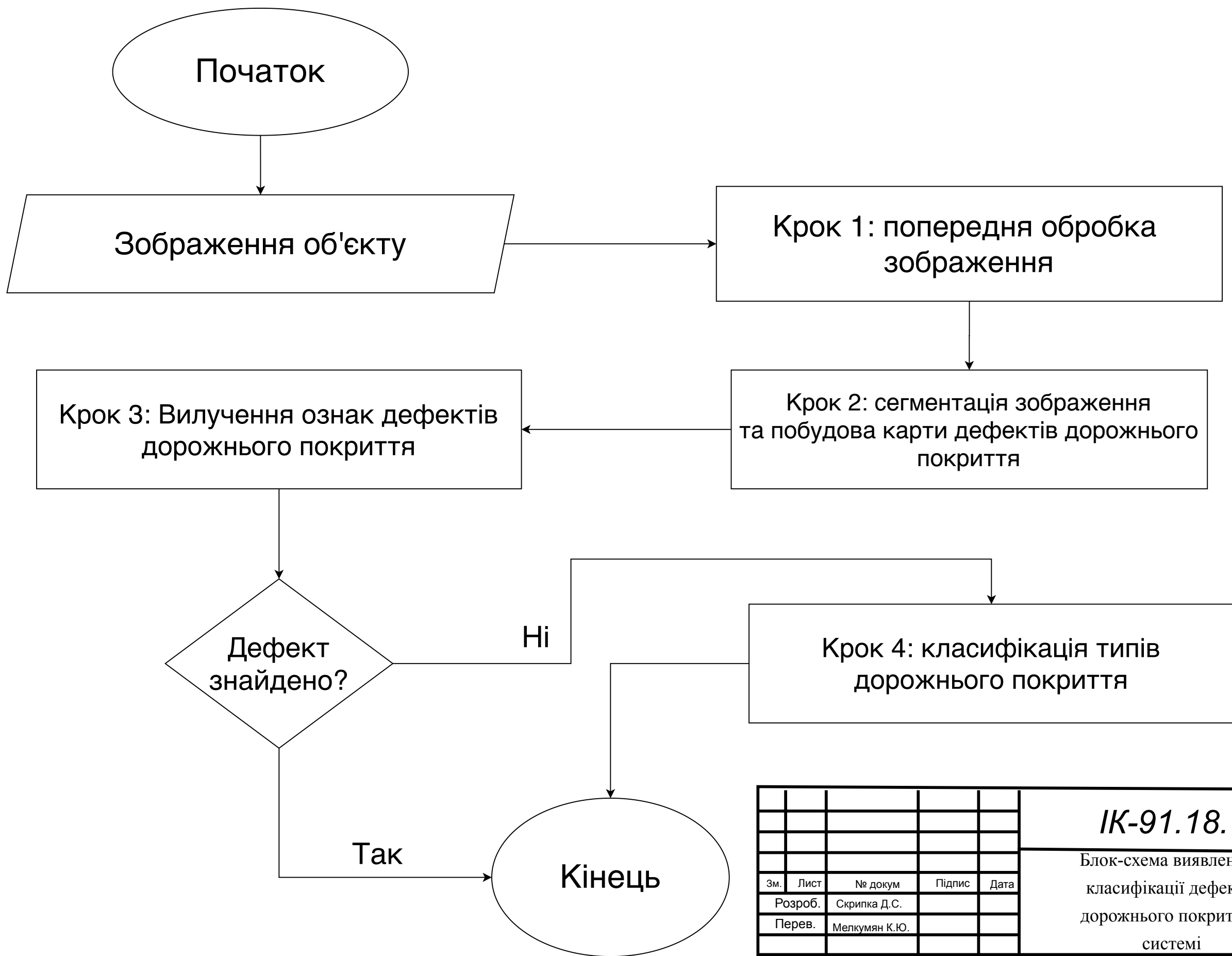


					ІК-91.18. 3132.007 ВЗ							
					Блок-схема сегментіції зображення	Літ.		Маса		Мірило		
Зм.	Лист	№ докум	Підпис	Дата								
Розроб.		Скрипка Д.С										
Перев.		Польшакова О.М										
					Кафедра Технічної кібернетики	Лист 1		Листів 1				
Н.контр	Пасько В.П								Група ІК-91мп			
Затв.		Пархомей І.Р.										



## ДОДАТОК Г

Блок – схема виявлення и класифікації дефектів дорожнього покриття в системі



Зм.	Лист	№ докум	Підпис	Дата
Розроб.		Скрипка Д.С.		
Перев.		Мелкумян К.Ю.		
Н.контр		Пасько В.П		
Затв.		Пархомей І.Р.		

ІК-91.18. 3132.003 ВЗ

Блок-схема виявлення і  
класифікації дефектів  
дорожнього покриття в  
системі

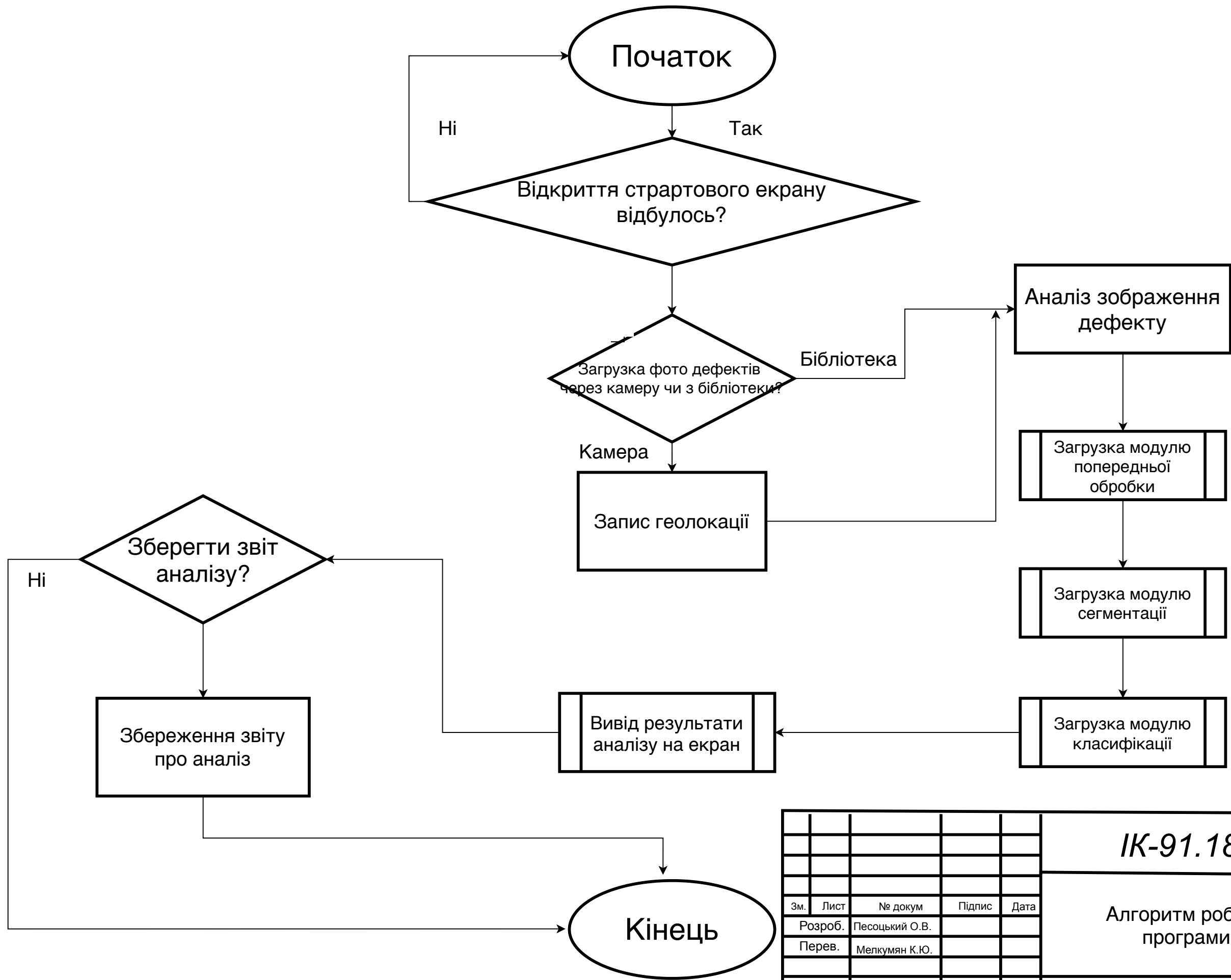
Кафедра  
Технічної кібернетики

Літ.	Маса	Мірило
Лист 1	Листів 1	

Група ІК-91мп

## ДОДАТОК Г

### Алгоритм роботи програми



Зм.	Лист	№ докум	Підпис	Дата
Розроб.		Песоцький О.В.		
Перев.		Мелкумян К.Ю.		
Н.контр		Пасько В.П		
Затв.		Пархомей І.Р.		

ІК-91.18. 3132.003 ВЗ

Алгоритм роботи  
програми

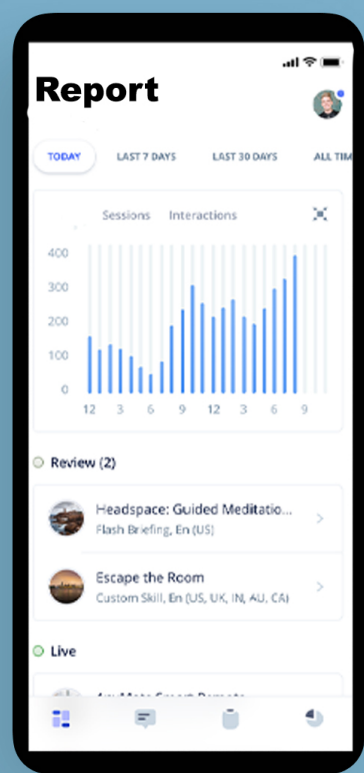
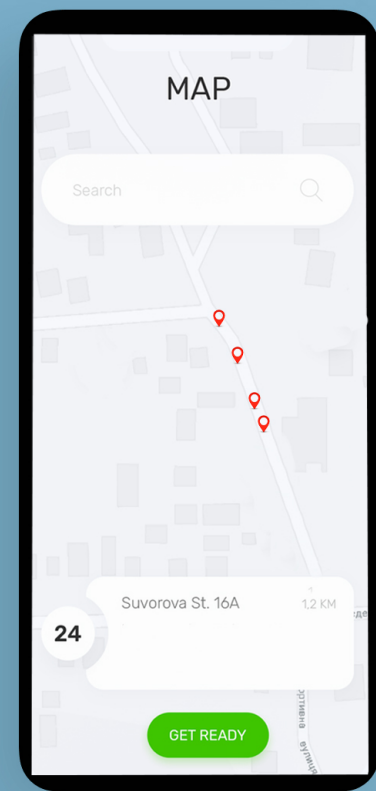
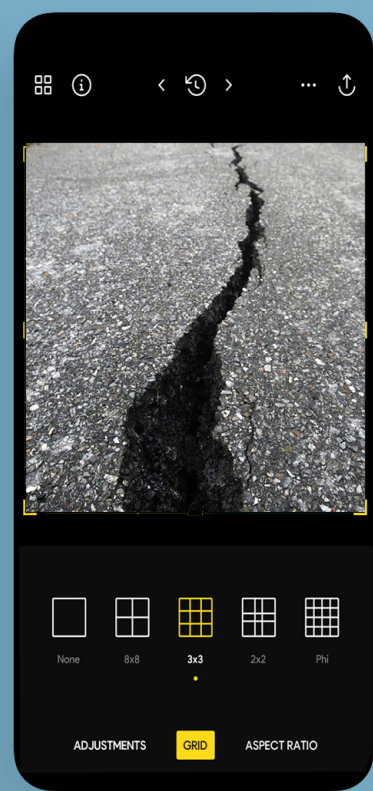
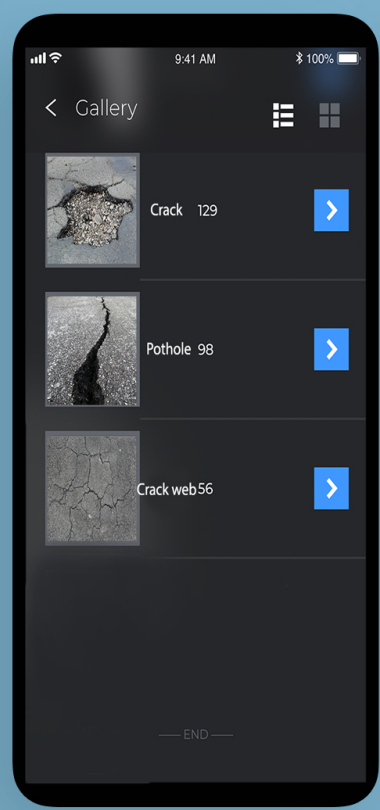
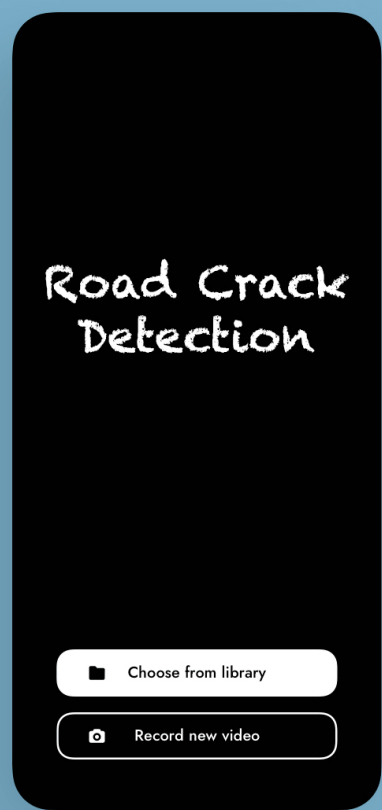
Кафедра  
Технічної кібернетики

Літ.	Маса	Мірило
Лист	1	Листів
		1

Група ІК-91мп

## ДОДАТОК Д

### Інтерфейс роботи програми



					ІК-91.18. 3132.003 ВЗ						
					Інтерфейс роботи програми	Літ.		Маса		Мірило	
Зм.	Лист	№ докум	Підпис	Дата		Лист		1	Листів		1
Розроб.		Песоцький О.В.									
Перев.		Мелкумян К.Ю.									
Н.контр		Пасько В.П			Кафедра Технічної кібернетики	Група ІК-91мп					
Затв.		Пархомей І.Р.									

## ДОДАТОК Е

### Лістинг програми

```
//
!$*UTF8*$!

{
    archiveVersion = 1;
    classes = {
    };
    objectVersion = 50;
    objects = {

/* Begin PBXBuildFile section */
        1172A50321F86BF3006F0BD4 /* AppDelegate.swift in Sources */ =
        {isa = PBXBuildFile; fileRef = 1172A50221F86BF3006F0BD4 /* AppDelegate.swift
        */; };
        1172A50521F86BF3006F0BD4 /* ViewController.swift in Sources */
        = {isa = PBXBuildFile; fileRef = 1172A50421F86BF3006F0BD4 /*
        ViewController.swift */; };
        1172A50821F86BF3006F0BD4 /* Main.storyboard in Resources */ =
        {isa = PBXBuildFile; fileRef = 1172A50621F86BF3006F0BD4 /* Main.storyboard */;
        };
        1172A50A21F86BF4006F0BD4 /* Assets.xcassets in Resources */ =
        {isa = PBXBuildFile; fileRef = 1172A50921F86BF4006F0BD4 /* Assets.xcassets */;
        };
        1172A50D21F86BF4006F0BD4 /* LaunchScreen.storyboard in
        Resources */ = {isa = PBXBuildFile; fileRef = 1172A50B21F86BF4006F0BD4 /*
        LaunchScreen.storyboard */; };
        1172A52321F88FCA006F0BD4 /* SimpleNN.framework in Frameworks */
        = {isa = PBXBuildFile; fileRef = 1172A52221F88FCA006F0BD4 /*
        SimpleNN.framework */; };
        1172A52421F88FCA006F0BD4 /* SimpleNN.framework in Embed
        Frameworks */ = {isa = PBXBuildFile; fileRef = 1172A52221F88FCA006F0BD4 /*
        SimpleNN.framework */; settings = {ATTRIBUTES = (CodeSignOnCopy,
        RemoveHeadersOnCopy, ); }; };
        1172A53921FABB15006F0BD4 /* pothole.jpg in Resources */ = {isa
        = PBXBuildFile; fileRef = 1172A53221FABB13006F0BD4 /* pothole.jpg */; };
        1172A53A21FABB15006F0BD4 /* crack.jpg in Resources */ = {isa =
        PBXBuildFile; fileRef = 1172A53321FABB13006F0BD4 /* crack.jpg */; };
        1172A53B21FABB15006F0BD4 /* crack web.jpg in Resources */ =
        {isa = PBXBuildFile; fileRef = 1172A53421FABB14006F0BD4 /* crack web.jpg */;
        };
/* End PBXBuildFile section */

/* Begin PBXCopyFilesBuildPhase section */
        1172A52521F88FCA006F0BD4 /* Embed Frameworks */ = {
            isa = PBXCopyFilesBuildPhase;
            buildActionMask = 2147483647;
            dstPath = "";
            dstSubfolderSpec = 10;
        };
    };
}
```



```

        files = (
            1172A52421F88FCA006F0BD4 /* SimpleNN.framework in
Embed Frameworks */,
        );
        name = "Embed Frameworks";
        runOnlyForDeploymentPostprocessing = 0;
    };
/* End PBXCopyFilesBuildPhase section */

/* Begin PBXFileReference section */
        1172A4FF21F86BF3006F0BD4 /* ObjectDetection.app */ = {isa =
PBXFileReference; explicitFileType = wrapper.application; includeInIndex = 0;
path = ObjectDetection.app; sourceTree = BUILT_PRODUCTS_DIR; };
        1172A50221F86BF3006F0BD4 /* AppDelegate.swift */ = {isa =
PBXFileReference; lastKnownFileType = sourcecode.swift; path =
AppDelegate.swift; sourceTree = "<group>"; };
        1172A50421F86BF3006F0BD4 /* ViewController.swift */ = {isa =
PBXFileReference; lastKnownFileType = sourcecode.swift; path =
ViewController.swift; sourceTree = "<group>"; };
        1172A50721F86BF3006F0BD4 /* Base */ = {isa = PBXFileReference;
lastKnownFileType = file.storyboard; name = Base; path =
Base.lproj/Main.storyboard; sourceTree = "<group>"; };
        1172A50921F86BF4006F0BD4 /* Assets.xcassets */ = {isa =
PBXFileReference; lastKnownFileType = folder.assetcatalog; path =
Assets.xcassets; sourceTree = "<group>"; };
        1172A50C21F86BF4006F0BD4 /* Base */ = {isa = PBXFileReference;
lastKnownFileType = file.storyboard; name = Base; path =
Base.lproj/LaunchScreen.storyboard; sourceTree = "<group>"; };
        1172A50E21F86BF4006F0BD4 /* Info.plist */ = {isa =
PBXFileReference; lastKnownFileType = text.plist.xml; path = Info.plist;
sourceTree = "<group>"; };
        1172A52221F88FCA006F0BD4 /* SimpleNN.framework */ = {isa =
PBXFileReference; explicitFileType = wrapper.framework; path =
SimpleNN.framework; sourceTree = BUILT_PRODUCTS_DIR; };
        1172A53221FABB13006F0BD4 /* pothole.jpg */ = {isa =
PBXFileReference; lastKnownFileType = image.jpeg; path = pothole.jpg;
sourceTree = "<group>"; };
        1172A53321FABB13006F0BD4 /* crack.jpg */ = {isa =
PBXFileReference; lastKnownFileType = image.jpeg; path = crack.jpg; sourceTree
= "<group>"; };
        1172A53421FABB14006F0BD4 /* crack web.jpg */ = {isa =
PBXFileReference; lastKnownFileType = image.jpeg; path = crack web.jpg;
sourceTree = "<group>"; };
/* End PBXFileReference section */

/* Begin PBXFrameworksBuildPhase section */
        1172A4FC21F86BF3006F0BD4 /* Frameworks */ = {

```

```

        isa = PBXFrameworksBuildPhase;
        buildActionMask = 2147483647;
        files = (
            1172A52321F88FCA006F0BD4 /* SimpleNN.framework in
Frameworks */,
        );

        runOnlyForDeploymentPostprocessing = 0;
    };

/* End PBXFrameworksBuildPhase section */

/* Begin PBXGroup section */
    1172A4F621F86BF3006F0BD4 = {
        isa = PBXGroup;
        children = (
            1172A52221F88FCA006F0BD4 /* SimpleNN.framework
*/,

            1172A50121F86BF3006F0BD4 /* ObjectDetection */,
            1172A50021F86BF3006F0BD4 /* Products */,

        );

        sourceTree = "<group>";
    };
    1172A50021F86BF3006F0BD4 /* Products */ = {
        isa = PBXGroup;
        children = (
            1172A4FF21F86BF3006F0BD4 /* ObjectDetection.app
*/,

        );
        name = Products;
        sourceTree = "<group>";
    };
    1172A50121F86BF3006F0BD4 /* ObjectDetection */ = {
        isa = PBXGroup;
        children = (
            1172A53621FABB14006F0BD4 /* pothole.jpg */,
            1172A53721FABB15006F0BD4 /* crack.jpg */,
            1172A53321FABB13006F0BD4 /* crack web.jpg */,
            1172A50221F86BF3006F0BD4 /* AppDelegate.swift */,
            1172A50421F86BF3006F0BD4 /* ViewController.swift
*/,

            1172A50621F86BF3006F0BD4 /* Main.storyboard */,
            1172A50921F86BF4006F0BD4 /* Assets.xcassets */,
            1172A50B21F86BF4006F0BD4 /*
LaunchScreen.storyboard */,

            1172A50E21F86BF4006F0BD4 /* Info.plist */,

        );
        path = ObjectDetection;
        sourceTree = "<group>";
    };

```

```

    };
/* End PBXGroup section */

/* Begin PBXNativeTarget section */
    1172A4FE21F86BF3006F0BD4 /* ObjectDetection */ = {
        isa = PBXNativeTarget;
        buildConfigurationList = 1172A51121F86BF4006F0BD4 /*
Build configuration list for PBXNativeTarget "ObjectDetection" */;
        buildPhases = (
            1172A4FB21F86BF3006F0BD4 /* Sources */,
            1172A4FC21F86BF3006F0BD4 /* Frameworks */,
            1172A4FD21F86BF3006F0BD4 /* Resources */,
            1172A52521F88FCA006F0BD4 /* Embed Frameworks */,
        );
        buildRules = (
            );
        dependencies = (
        );
        name = ObjectDetection;
        productName = ObjectDetection;
        productReference = 1172A4FF21F86BF3006F0BD4 /*
ObjectDetection.app */;
        productType = "com.apple.product-type.application";
    };
/* End PBXNativeTarget section */

/* Begin PBXProject section */
    1172A4F721F86BF3006F0BD4 /* Project object */ = {
        isa = PBXProject;
        attributes = {
            LastSwiftUpdateCheck = 1010;
            LastUpgradeCheck = 1010;
            ORGANIZATIONNAME = SimpleTech;
            TargetAttributes = {
                1172A4FE21F86BF3006F0BD4 = {
                    CreatedOnToolsVersion = 10.1;
                };
            };
        };
        buildConfigurationList = 1172A4FA21F86BF3006F0BD4 /*
Build configuration list for PBXProject "ObjectDetection" */;
        compatibilityVersion = "Xcode 9.3";
        developmentRegion = en;
        hasScannedForEncodings = 0;
        knownRegions = (
            en,
            Base,

```

```

    );
    mainGroup = 1172A4F621F86BF3006F0BD4;
    productRefGroup = 1172A50021F86BF3006F0BD4 /* Products
*/;

    projectDirPath = "";
    projectRoot = "";
    targets = (
        1172A4FE21F86BF3006F0BD4 /* ObjectDetection */,
    );
};
/* End PBXProject section */

```

Наступним модулем розробки виступав модуль попередньої обробки зображення :

```

/* Begin
XCBuildConfiguration
section */

1172A50F21F86BF4006F0BD4 /* Debug */ = {
    isa = XCBuildConfiguration;
    buildSettings = {
        ALWAYS_SEARCH_USER_PATHS = NO;
        CLANG_ANALYZER_NONNULL = YES;
        CLANG_ANALYZER_NUMBER_OBJECT_CONVERSION
= YES_AGGRESSIVE;
        CLANG_CXX_LANGUAGE_STANDARD =
"gnu++14";
        CLANG_CXX_LIBRARY = "libc++";
        CLANG_ENABLE_MODULES = YES;
        CLANG_ENABLE_OBJC_ARC = YES;
        CLANG_ENABLE_OBJC_WEAK = YES;
        CLANG_WARN_BLOCK_CAPTURE_AUTORELEASING
= YES;
        CLANG_WARN_BOOL_CONVERSION = YES;
        CLANG_WARN_COMMA = YES;
        CLANG_WARN_CONSTANT_CONVERSION = YES;
        CLANG_WARN_DEPRECATED_OBJC_IMPLEMENTATIONS = YES;
        CLANG_WARN_DIRECT_OBJC_ISA_USAGE =
YES_ERROR;
        CLANG_WARN_DOCUMENTATION_COMMENTS =
YES;
        CLANG_WARN_EMPTY_BODY = YES;
        CLANG_WARN_ENUM_CONVERSION = YES;

```

```

CLANG_WARN_INFINITE_RECURSION = YES;
CLANG_WARN_INT_CONVERSION = YES;
CLANG_WARN_NON_LITERAL_NULL_CONVERSION
= YES;

CLANG_WARN_OBJC_IMPLICIT_RETAIN_SELF =
YES;

CLANG_WARN_OBJC_LITERAL_CONVERSION =
YES;

CLANG_WARN_OBJC_ROOT_CLASS = YES_ERROR;
CLANG_WARN_RANGE_LOOP_ANALYSIS = YES;
CLANG_WARN_STRICT_PROTOTYPES = YES;
CLANG_WARN_SUSPICIOUS_MOVE = YES;
CLANG_WARN_UNGUARDED_AVAILABILITY =
YES_AGGRESSIVE;

CLANG_WARN_UNREACHABLE_CODE = YES;
CLANG_WARN__DUPLICATE_METHOD_MATCH =
YES;

CODE_SIGN_IDENTITY = "iPhone
Developer";

COPY_PHASE_STRIP = NO;
DEBUG_INFORMATION_FORMAT = dwarf;
ENABLE_STRICT_OBJC_MSGSEND = YES;
ENABLE_TESTABILITY = YES;
GCC_C_LANGUAGE_STANDARD = gnu11;
GCC_DYNAMIC_NO_PIC = NO;
GCC_NO_COMMON_BLOCKS = YES;
GCC_OPTIMIZATION_LEVEL = 0;
GCC_PREPROCESSOR_DEFINITIONS = (
    "DEBUG=1",
    "$(inherited)",
);
GCC_WARN_64_TO_32_BIT_CONVERSION = YES;
GCC_WARN_ABOUT_RETURN_TYPE = YES_ERROR;
GCC_WARN_UNDECLARED_SELECTOR = YES;
GCC_WARN_UNINITIALIZED_AUTOS =
YES_AGGRESSIVE;

GCC_WARN_UNUSED_FUNCTION = YES;
GCC_WARN_UNUSED_VARIABLE = YES;
IPHONEOS_DEPLOYMENT_TARGET = 12.1;
MTL_ENABLE_DEBUG_INFO = INCLUDE_SOURCE;
MTL_FAST_MATH = YES;
ONLY_ACTIVE_ARCH = YES;
SDKROOT = iphoneos;
SWIFT_ACTIVE_COMPILATION_CONDITIONS =
DEBUG;

SWIFT_OPTIMIZATION_LEVEL = "-Onone";
};

```

```

        name = Debug;
    };
    1172A51021F86BF4006F0BD4 /* Release */ = {
        isa = XCBuildConfiguration;
        buildSettings = {
            ALWAYS_SEARCH_USER_PATHS = NO;
            CLANG_ANALYZER_NONNULL = YES;
            CLANG_ANALYZER_NUMBER_OBJECT_CONVERSION
= YES_AGGRESSIVE;
            CLANG_CXX_LANGUAGE_STANDARD =
"gnu++14";
            CLANG_CXX_LIBRARY = "libc++";
            CLANG_ENABLE_MODULES = YES;
            CLANG_ENABLE_OBJC_ARC = YES;
            CLANG_ENABLE_OBJC_WEAK = YES;
            CLANG_WARN_BLOCK_CAPTURE_AUTORELEASING
= YES;
            CLANG_WARN_BOOL_CONVERSION = YES;
            CLANG_WARN_COMMA = YES;
            CLANG_WARN_CONSTANT_CONVERSION = YES;
            CLANG_WARN_DEPRECATED_OBJC_IMPLEMENTATIONS = YES;
            CLANG_WARN_DIRECT_OBJC_ISA_USAGE =
YES_ERROR;
            CLANG_WARN_DOCUMENTATION_COMMENTS =
YES;
            CLANG_WARN_EMPTY_BODY = YES;
            CLANG_WARN_ENUM_CONVERSION = YES;
            CLANG_WARN_INFINITE_RECURSION = YES;
            CLANG_WARN_INT_CONVERSION = YES;
            CLANG_WARN_NON_LITERAL_NULL_CONVERSION
= YES;
            CLANG_WARN_OBJC_IMPLICIT_RETAIN_SELF =
YES;
            CLANG_WARN_OBJC_LITERAL_CONVERSION =
YES;
            CLANG_WARN_OBJC_ROOT_CLASS = YES_ERROR;
            CLANG_WARN_RANGE_LOOP_ANALYSIS = YES;
            CLANG_WARN_STRICT_PROTOTYPES = YES;
            CLANG_WARN_SUSPICIOUS_MOVE = YES;
            CLANG_WARN_UNGUARDED_AVAILABILITY =
YES_AGGRESSIVE;
            CLANG_WARN_UNREACHABLE_CODE = YES;
            CLANG_WARN__DUPLICATE_METHOD_MATCH =
YES;
            CODE_SIGN_IDENTITY = "iPhone
Developer";

```

```

COPY_PHASE_STRIP = NO;
DEBUG_INFORMATION_FORMAT = "dwarf-with-
dsym";

ENABLE_NS_ASSERTIONS = NO;
ENABLE_STRICT_OBJC_MSGSEND = YES;
GCC_C_LANGUAGE_STANDARD = gnu11;
GCC_NO_COMMON_BLOCKS = YES;
GCC_WARN_64_TO_32_BIT_CONVERSION = YES;
GCC_WARN_ABOUT_RETURN_TYPE = YES_ERROR;
GCC_WARN_UNDECLARED_SELECTOR = YES;
GCC_WARN_UNINITIALIZED_AUTOS =
YES_AGGRESSIVE;

GCC_WARN_UNUSED_FUNCTION = YES;
GCC_WARN_UNUSED_VARIABLE = YES;
IPHONEOS_DEPLOYMENT_TARGET = 12.1;
MTL_ENABLE_DEBUG_INFO = NO;
MTL_FAST_MATH = YES;
SDKROOT = iphoneos;
SWIFT_COMPILATION_MODE = wholemodule;
SWIFT_OPTIMIZATION_LEVEL = "-O";
VALIDATE_PRODUCT = YES;
};
name = Release;
};
1172A51221F86BF4006F0BD4 /* Debug */ = {
    isa = XCBuildConfiguration;
    buildSettings = {
        ASSETCATALOG_COMPILER_APPICON_NAME =
AppIcon;

        CODE_SIGN_STYLE = Automatic;
        DEVELOPMENT_TEAM = U7KXBCD2JV;
        INFOPLIST_FILE =
ObjectDetection/Info.plist;

        IPHONEOS_DEPLOYMENT_TARGET = 10.0;
        LD_RUNPATH_SEARCH_PATHS = (
            "${inherited}",
            "@executable_path/Frameworks",
        );
        PRODUCT_BUNDLE_IDENTIFIER =
com.st.ObjectDetection;

        PRODUCT_NAME = "${TARGET_NAME}";
        SWIFT_VERSION = 4.2;
        TARGETED_DEVICE_FAMILY = "1,2";
    };
    name = Debug;
};
1172A51321F86BF4006F0BD4 /* Release */ = {

```

```

        isa = XCBuildConfiguration;
        buildSettings = {
            ASSETCATALOG_COMPILER_APPICON_NAME =
AppIcon;

            CODE_SIGN_STYLE = Automatic;
            DEVELOPMENT_TEAM = U7KXBCD2JV;
            INFOPLIST_FILE =
ObjectDetection/Info.plist;

            IPHONEOS_DEPLOYMENT_TARGET = 10.0;
            LD_RUNPATH_SEARCH_PATHS = (
                "${inherited}",
                "@executable_path/Frameworks",
            );
            PRODUCT_BUNDLE_IDENTIFIER =
com.st.ObjectDetection;

            PRODUCT_NAME = "${TARGET_NAME}";
            SWIFT_VERSION = 4.2;
            TARGETED_DEVICE_FAMILY = "1,2";
        };
        name = Release;
    };

/* End XCBuildConfiguration section */

/* Begin XCConfigurationList section */
1172A4FA21F86BF3006F0BD4 /* Build configuration list
for PBXProject "ObjectDetection" */ = {
    isa = XCConfigurationList;
    buildConfigurations = (
        1172A50F21F86BF4006F0BD4 /* Debug */,
        1172A51021F86BF4006F0BD4 /* Release */,
    );
    defaultConfigurationIsVisible = 0;
    defaultConfigurationName = Release;
};
1172A51121F86BF4006F0BD4 /* Build configuration list
for PBXNativeTarget "ObjectDetection" */ = {
    isa = XCConfigurationList;
    buildConfigurations = (
        1172A51221F86BF4006F0BD4 /* Debug */,
        1172A51321F86BF4006F0BD4 /* Release */,
    );
    defaultConfigurationIsVisible = 0;
    defaultConfigurationName = Release;
};

/* End XCConfigurationList section */
};

rootObject = 1172A4F721F86BF3006F0BD4 /* Project object */;

```



```
}
```

Після чого було розроблено модуль класифікації :

```
{
    archiveVersion = 1;
    classes = {
    };
    objectVersion = 50;
    objects = {

/* Begin PBXBuildFile section */
        113FE05B21EF68C70030CD12 /* AppDelegate.swift in Sources */ = {isa =
PBXBuildFile; fileRef = 113FE05A21EF68C70030CD12 /* AppDelegate.swift */; };
        113FE05D21EF68C70030CD12 /* ViewController.swift in Sources */ = {isa
= PBXBuildFile; fileRef = 113FE05C21EF68C70030CD12 /* ViewController.swift */; };
        113FE06021EF68C70030CD12 /* Main.storyboard in Resources */ = {isa =
PBXBuildFile; fileRef = 113FE05E21EF68C70030CD12 /* Main.storyboard */; };
        113FE06221EF68C80030CD12 /* Assets.xcassets in Resources */ = {isa =
PBXBuildFile; fileRef = 113FE06121EF68C80030CD12 /* Assets.xcassets */; };
        113FE06521EF68C80030CD12 /* LaunchScreen.storyboard in Resources */ =
{isa = PBXBuildFile; fileRef = 113FE06321EF68C80030CD12 /* LaunchScreen.storyboard
*/; };
        113FE06D21EF69900030CD12 /* SimpleNN.framework in Frameworks */ =
{isa = PBXBuildFile; fileRef = 113FE06C21EF69900030CD12 /* SimpleNN.framework */;
};
        113FE06E21EF69900030CD12 /* SimpleNN.framework in Embed Frameworks */
= {isa = PBXBuildFile; fileRef = 113FE06C21EF69900030CD12 /* SimpleNN.framework */;
settings = {ATTRIBUTES = (CodeSignOnCopy, RemoveHeadersOnCopy, ); }; };
        113FE07321EF6BB20030CD12 /* mobilenet.bin in Resources */ = {isa =
PBXBuildFile; fileRef = 113FE07121EF6BB20030CD12 /* mobilenet.bin */; };
        113FE07421EF6BB20030CD12 /* mobilenet.json in Resources */ = {isa =
PBXBuildFile; fileRef = 113FE07221EF6BB20030CD12 /* mobilenet.json */; };
        113FE07621EF6BD00030CD12 /* crack.png in Resources */ = {isa =
PBXBuildFile; fileRef = 113FE07521EF6BD00030CD12 /* crack.png */; };
        113FE07821EF6C080030CD12 /* synset_words.txt in Resources */ = {isa =
PBXBuildFile; fileRef = 113FE07721EF6C080030CD12 /* synset_words.txt */; };
/* End PBXBuildFile section */

/* Begin PBXCopyFilesBuildPhase section */
        113FE06F21EF69910030CD12 /* Embed Frameworks */ = {
            isa = PBXCopyFilesBuildPhase;
            buildActionMask = 2147483647;
            dstPath = "";
            dstSubfolderSpec = 10;
            files = (
                113FE06E21EF69900030CD12 /* SimpleNN.framework in
Embed Frameworks */,
            );
        };
    };
}
```

```

        name = "Embed Frameworks";
        runOnlyForDeploymentPostprocessing = 0;
    };
/* End PBXCopyFilesBuildPhase section */

/* Begin PBXFileReference section */
    113FE05721EF68C70030CD12 /* ImageClassification.app */ = {isa =
PBXFileReference; explicitFileType = wrapper.application; includeInIndex = 0; path
= ImageClassification.app; sourceTree = BUILT_PRODUCTS_DIR; };
    113FE05A21EF68C70030CD12 /* AppDelegate.swift */ = {isa =
PBXFileReference; lastKnownFileType = sourcecode.swift; path = AppDelegate.swift;
sourceTree = "<group>"; };
    113FE05C21EF68C70030CD12 /* ViewController.swift */ = {isa =
PBXFileReference; lastKnownFileType = sourcecode.swift; path =
ViewController.swift; sourceTree = "<group>"; };
    113FE05F21EF68C70030CD12 /* Base */ = {isa = PBXFileReference;
lastKnownFileType = file.storyboard; name = Base; path =
Base.lproj/Main.storyboard; sourceTree = "<group>"; };
    113FE06121EF68C80030CD12 /* Assets.xcassets */ = {isa =
PBXFileReference; lastKnownFileType = folder.assetcatalog; path = Assets.xcassets;
sourceTree = "<group>"; };
    113FE06421EF68C80030CD12 /* Base */ = {isa = PBXFileReference;
lastKnownFileType = file.storyboard; name = Base; path =
Base.lproj/LaunchScreen.storyboard; sourceTree = "<group>"; };
    113FE06621EF68C80030CD12 /* Info.plist */ = {isa = PBXFileReference;
lastKnownFileType = text.plist.xml; path = Info.plist; sourceTree = "<group>"; };
    113FE06C21EF69900030CD12 /* SimpleNN.framework */ = {isa =
PBXFileReference; explicitFileType = wrapper.framework; path = SimpleNN.framework;
sourceTree = BUILT_PRODUCTS_DIR; };
    113FE07121EF6BB20030CD12 /* mobilenet.bin */ = {isa =
PBXFileReference; lastKnownFileType = archive.macbinary; name = mobilenet.bin; path
= ImageClassification/network/mobilenet.bin; sourceTree = SOURCE_ROOT; };
    113FE07221EF6BB20030CD12 /* mobilenet.json */ = {isa =
PBXFileReference; fileEncoding = 4; lastKnownFileType = text.json; name =
mobilenet.json; path = ImageClassification/network/mobilenet.json; sourceTree =
SOURCE_ROOT; };
    113FE07521EF6BD00030CD12 /* cat.png */ = {isa = PBXFileReference;
lastKnownFileType = image.png; path = cat.png; sourceTree = "<group>"; };
    113FE07721EF6C080030CD12 /* synset_words.txt */ = {isa =
PBXFileReference; fileEncoding = 4; lastKnownFileType = text; path =
synset_words.txt; sourceTree = "<group>"; };
/* End PBXFileReference section */

/* Begin PBXFrameworksBuildPhase section */
    113FE05421EF68C70030CD12 /* Frameworks */ = {
        isa = PBXFrameworksBuildPhase;
        buildActionMask = 2147483647;
    };

```

```

        files = (
            113FE06D21EF69900030CD12 /* SimpleNN.framework in
Frameworks */,
        );
        runOnlyForDeploymentPostprocessing = 0;
    };
/* End PBXFrameworksBuildPhase section */

/* Begin PBXGroup section */
    113FE04E21EF68C70030CD12 = {
        isa = PBXGroup;
        children = (
            113FE06C21EF69900030CD12 /* SimpleNN.framework */,
            113FE05921EF68C70030CD12 /* ImageClassification */,
            113FE05821EF68C70030CD12 /* Products */,
        );
        sourceTree = "<group>";
    };
    113FE05821EF68C70030CD12 /* Products */ = {
        isa = PBXGroup;
        children = (
            113FE05721EF68C70030CD12 /* ImageClassification.app */,
        );
        name = Products;
        sourceTree = "<group>";
    };
    113FE05921EF68C70030CD12 /* ImageClassification */ = {
        isa = PBXGroup;
        children = (
            113FE07721EF6C080030CD12 /* synset_words.txt */,
            113FE07521EF6BD00030CD12 /* crack.png */,
            113FE07021EF6BA50030CD12 /* network */,
            113FE05A21EF68C70030CD12 /* AppDelegate.swift */,
            113FE05C21EF68C70030CD12 /* ViewController.swift */,
            113FE05E21EF68C70030CD12 /* Main.storyboard */,
            113FE06121EF68C80030CD12 /* Assets.xcassets */,
            113FE06321EF68C80030CD12 /* LaunchScreen.storyboard
*/,
            113FE06621EF68C80030CD12 /* Info.plist */,
        );
        path = ImageClassification;
        sourceTree = "<group>";
    };
    113FE07021EF6BA50030CD12 /* network */ = {
        isa = PBXGroup;
        children = (
            113FE07121EF6BB20030CD12 /* mobilenet.bin */,

```

```
113FE07221EF6BB20030CD12 /* mobilenet.json */,  
);  
name = network;  
path = net;  
sourceTree = "<group>";  
};  
/* End PBXGroup section */
```

## ДОДАТОК Є

### Результати перевірки на співпадіння

Ім'я користувача:  
Лісовиченко Олег Іванович

ID перевірки:  
1005463054

Дата перевірки:  
15.12.2020 15:56:11 EET

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
15.12.2020 18:03:22 EET

ID користувача:  
76913

Назва документа: Скрипка Д.С. ІК-91мп

Кількість сторінок: 46 Кількість слів: 8913 Кількість символів: 70524 Розмір файлу: 233.95 KB ID файлу: 1005752802

## 5.6% Схожість

Найбільша схожість: 1.05% з джерелом з Бібліотеки (ID файлу: 1005645136)

2.28% Джерела з Інтернету 85 ..... Сторінка 48

5.19% Джерела з Бібліотеки 163 ..... Сторінка 48

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 55